

53-1002434-01
January, 2012



ServerIron ADX

Administration Guide

Supporting Brocade ServerIron ADX version 12.4.00

BROCADE

© 2012 Brocade Communications Systems, Inc. All Rights Reserved.

Brocade, Brocade Assurance, the B-wing symbol, DCX, Fabric OS, MLX, SAN Health, VCS, and VDX are registered trademarks, and AnyIO, Brocade One, CloudPlex, Effortless Networking, ICX, NET Health, OpenScript, and The Effortless Network are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. Other brands, products, or service names mentioned may be trademarks of their respective owners.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. shall have no liability or responsibility to any person or entity with respect to any loss, cost, liability, or damages arising from the information contained in this book or the computer programs that accompany it.

The product described by this document may contain "open source" software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit

<http://www.brocade.com/support/oscd>.

Brocade Communications Systems, Incorporated

Corporate and Latin American Headquarters
Brocade Communications Systems, Inc.
130 Holger Way
San Jose, CA 95134
Tel: 1-408-333-8000
Fax: 1-408-333-8101
E-mail: info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems China HK, Ltd.
No. 1 Guanghua Road
Chao Yang District
Units 2718 and 2818
Beijing 100020, China
Tel: +8610 6588 8888
Fax: +8610 6588 9999
E-mail: china-info@brocade.com

European Headquarters
Brocade Communications Switzerland Sàrl
Centre Swissair
Tour B - 4ème étage
29, Route de l'Aéroport
Case Postale 105
CH-1215 Genève 15
Switzerland
Tel: +41 22 799 5640
Fax: +41 22 799 5641
E-mail: emea-info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems Co., Ltd. (Shenzhen WFOE)
Citic Plaza
No. 233 Tian He Road North
Unit 1308 – 13th Floor
Guangzhou, China
Tel: +8620 3891 2000
Fax: +8620 3891 2111
E-mail: china-info@brocade.com

Document History

Title	Publication number	Summary of changes	Date
<i>ServerIron ADX Administration Guide</i>	<i>53-1002434-01</i>	New document	January, 2012

Contents

About This Document

In this chapter	xiii
Audience	xiii
Supported hardware and software	xiii
Document conventions	xiii
Text formatting	xiii
Notes, cautions, and danger notices	xiv
Notice to the reader	xiv
Related publications	xv
Getting technical help or reporting errors	xv
Web access	xv
E-mail and telephone access	xv

Chapter 1

ServerIron System Management

Setting up local user accounts	1
Configuring Telnet	2
Enabling Telnet authentication	3
Enabling Telnet password	3
Using a standard ACL to control Telnet access	3
Restricting Telnet management access	4
Changing the Telnet login timeout period	4
Enabling or disabling Telnet access	4
Allowing Telnet access only to clients in a VLAN	4
Suppressing the rejection message	4
Defining Telnet timeout	5

Configuring SSH	6
Enabling or disabling SSH service	6
Creating a seed for generating a random number	6
Setting SSH authentication retries	6
Setting the SSH key size	7
Configuring SSH password authentication	7
Enabling empty password logins	7
Changing the TCP port used for SSH	8
Loading a public key file	8
Disabling or re-enabling RSA challenge-response authentication	8
Disabling or re-enabling Secure Copy	9
Using Secure Copy	9
Defining the SSH timeout value	10
Using a standard ACL to control SSH access	10
Displaying SSH information	10
Displaying currently loaded public keys	11
Managing System Functions	11
Terminating the active CLI session	11
Performing a lookup on a domain	11
Verifying connectivity	12
Tracing the IP path to a host	13
Initiating a system reset	13
Logging into a BP	14
Timing out idle serial management sessions	14
Configuring a ServerIron ADX to broadcast a session delete message	15
Assigning a name to the ServerIron ADX	15
Assigning an administrative ID	15
Disabling or re-enabling password encryption	15
Understanding dynamic configuration	16
Disabling or re-enabling the page-display mode	16
Disabling or re-enabling the stop page display characteristic	16
Configuring a message for display at the Privileged EXEC level	17
Configuring a message for display on a Console	17
Configuring a message for display on a terminal	17
Configuring TFTP	18
Using the management port	18
Configuring the management port	19
Using the USB port and USB flash drive	22
Clearing persistent information before an RMA	30
Erasing flash data	30
Clearing crash dump Information	30
Configuring SNTP	30
Configuring an SNTP server location	31
Defining how often the clock references are validated	31
Synchronizing the system clock	31
Displaying SNTP information	31

Configuring DNS	32
Defining a domain name	32
Defining DNS servers	33
Configuring DNS Resolver	33
Configuring SNMP	33
SNMP support	33
Traps	33
Using the MIB table	35
Restricting SNMP management access	35
Assigning an SNMP community string	36
Designating a contact	36
Enabling or disabling traps	36
Allowing SNMP access only to clients in a VLAN	37
Enabling or disabling a station as an SNMP trap receiver	37
Identifying a system location	37
Disabling password checking	37
Specifying the source for all SNMP traps	37
Configuring an SNMP view	38
Clearing all statistics for SNMP server traffic	38
Configuring access control	39
Enabling configuration of RADIUS	39
Enabling configuration of TACACS or TACACS+	39
Restricting management access to the ServerIronADX	39
Determining the access points where the password can be defined	39
Enhancing access privileges	40
TACACS and TACACS+	41
Setting TACACS or TACACS+ parameters	41
Enabling command authorization and accounting at the console	41
Displaying information about TACACS+ and RADIUS servers	42
RADIUS security	42
Setting RADIUS server parameters	42
Password recovery	43
Displaying information about the security feature	43
Power budgeting on the ServerIron ADX	44
Operation of power budgeting	44
Configuring the cooling system	45
Configuring a redundant management module	47
Synchronizing the active and standby modules	47
High availability configurations	48
Synchronizing the configurations	48
Preparing for synchronization	49
Initiating and ending the synchronization	49
Creating config-sync peers	50
Initiating the synchronization	51
Block-by-block synchronization	52

Displaying system information	54
Displaying and saving tech support information	57
Displaying statistics	61
Displaying port statistics	63
Displaying STP statistics	63
Displaying trunk group information	63
Clearing the statistics	63
Clearing all sessions	64
Using Syslog	64
Severity levels	64
Configuring logging	65
Displaying log information	66
Clearing syslog entries	69
Message format	69
Event logging	78
Disabling and enabling event logging	78
Changing the default maximum size of the event log	78
Displaying event log status	79
Clearing the event log	79
Additional system management functions	79
Configuring uplink utilization lists	79
Displaying an uplink utilization list	80
Setting system time and date	80
Activating or deactivating daylight savings time	81
Setting the time zone	81
DST change notice for networks using US time zones	81
Changing the shutdown temperature	82
Changing the temperature warning	82
Changing the number of seconds between polls	83
Disabling or re-enabling status polling	83
Adjusting inter-packet gap	83
Modifying the IPG on 10Mbps Ethernet segment	83
Modifying the IPG on 100Mbps Ethernet segment	84
Modifying the IPG on 1000Mbps Gigabit Ethernet segment	84
Assigning a port name	85
Modifying port speed and duplex mode	85
Enabling a mirror port	85
Displaying port mirroring and monitoring information	86
Setting QoS priority	86
Turning the flow control on or off	86
Setting the negotiation mode	87
Defining the performance mode	87
Forwarding Layer 2 and Layer 3 pass-through traffic to the CPU87	
Hardware forwarding for non L4-7 traffic flows	88
Remapping processing for a forwarding module to a BP	88
Specifying the maximum number of unknown unicast packets	89

Chapter 2

Secure Access Management

Securing access methods	91
-----------------------------------	----

Restricting remote access to management functions	92
Using ACLs to restrict remote access	93
Restricting remote access to the device to specific IP addresses	95
Restricting remote access to the device to specific VLAN IDs .	96
Designated VLAN for Telnet management sessions to a Layer 2	
Switch	97
Disabling specific access methods.	98
Setting passwords.	99
Setting a Telnet password	100
Setting passwords for management privilege levels	100
Recovering from a lost password	102
Displaying the SNMP community string	103
Disabling password encryption	104
Specifying a minimum password length.	104
Setting up local user accounts.	104
Configuring a local user account	105
Configuring TACACS or TACACS+ security	106
How TACACS+ differs from TACACS.	106
TACACS or TACACS+ authentication, authorization, and accounting	107
TACACS or TACACS+ configuration considerations	110
Identifying the TACACS or TACACS+ servers.	111
Specifying different servers for individual AAA functions . . .	112
Setting optional TACACS or TACACS+ parameters	112
Configuring authentication-method lists for TACACS or TACACS+	114
Configuring TACACS+ authorization	116
Configuring TACACS+ accounting	119
Configuring an interface as the source for all TACACS or TACACS+	120
packets	120
Displaying TACACS or TACACS+ statistics and configuration	
information.	121
Configuring RADIUS security	122
RADIUS authentication, authorization, and accounting	122
RADIUS NAS-Identifier	125
RADIUS configuration considerations.	125
RADIUS configuration procedure	126
Configuring Brocade-specific attributes on the RADIUS server	126
Identifying the RADIUS server to the ServerIron	127
Specifying different servers for individual AAA functions . . .	128
Setting RADIUS parameters	128
Configuring authentication-method lists for RADIUS.	129
Configuring RADIUS authorization	131
Configuring RADIUS accounting	133
Configuring an interface as the source for all RADIUS packets	134
Displaying RADIUS configuration information	135
Configuring authentication-method lists	136
Configuration considerations for authentication-method lists	137
Examples of authentication-method lists.	138

Chapter 3	Role Based Management	
	Overview	141
	Command Line Interface	143
	Integrating RBM with RADIUS and TACACS+	144
	Configuring the ServerIron ADX for AAA authentication	144
	Configuring the AAA server to authenticate RBM on a ServerIron ADX	146
Chapter 4	Securing SNMP Access	
	Establishing SNMP community strings	147
	Encryption of SNMP community strings	147
	Adding an SNMP community string	147
	Displaying the SNMP community strings	148
	Using the user-based security mode	149
	Configuring your NMS	149
	Configuring SNMP version 3 on the ServerIron ADX	149
	Defining the engine ID	150
	Defining an SNMP group	150
	Defining an SNMP user account	151
	Displaying the engine ID	153
	Displaying SNMP groups	153
	Displaying user information	154
	Interpreting varbinds in report packets	154
	SNMP v3 configuration examples	154
Chapter 5	Capacity on Demand	
	Software license terminology	157
	Software-based licensing overview	158
	License types	159
	Licensed features and SKU numbers	159
	Licensing rules	164
	Configuration tasks	165
	Obtaining a license	165
	Installing a license file	169
	Verifying the license file installation	170
	Using a trial license	170
	What happens when a trial license expires	171
	Console, and Syslog messages for trial license expiration	171
	Renewing or extending a trial license	171
	Deleting a license	171
	Viewing software license information from the Brocade software portal	172
	Syslog messages information	173

Viewing information about software licenses	174
Viewing the License ID (LID)	174
Viewing the license database	176

About This Document

In this chapter

• Audience.	xiii
• Supported hardware and software.	xiii
• Document conventions	xiii
• Notice to the reader	xiv
• Related publications	xv
• Getting technical help or reporting errors	xv

Audience

This guide is intended for network engineers with a basic knowledge of switching, routing, and application traffic management.

Supported hardware and software

Although many different software and hardware configurations are tested and supported by Brocade Communications Systems, Inc. for 12.3 documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release of this guide:

- ServerIron ADX 1000
- ServerIron ADX 4000
- ServerIron ADX 8000
- ServerIron ADX 10000

Document conventions

This section describes text formatting conventions and important notice formats used in this document.

Text formatting

The narrative-text formatting conventions that are used are as follows:

In this chapter

bold text	Identifies command names
	Identifies the names of user-manipulated GUI elements
	Identifies keywords
	Identifies text to enter at the GUI or CLI
<i>italic text</i>	Provides emphasis
	Identifies variables
	Identifies document titles
<code>code text</code>	Identifies CLI output

For readability, command names in the narrative portions of this guide are presented in bold: for example, **show version**.

Notes, cautions, and danger notices

The following notices and statements are used in this manual. They are listed below in order of increasing severity of potential hazards.

NOTE

A note provides a tip, guidance or advice, emphasizes important information, or provides a reference to related information.



CAUTION

A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.



DANGER

A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Notice to the reader

This document may contain references to the trademarks of the following corporations. These trademarks are the properties of their respective companies and corporations.

These references are made for informational purposes only.

Corporation	Referenced Trademarks and Products
Microsoft Corporation	Windows
The Open Group	UNIX

Related publications

The following Brocade documents supplement the information in this guide:

- *Release Notes for ServerIron Switch and Router Software TrafficWorks 12.3*
- *ServerIron ADX Graphical User Interface .*
- *ServerIron ADX Server Load Balancing Guide*
- *ServerIron ADX Advanced Server Load Balancing Guide*
- *ServerIron ADX Global Server Load Balancing Guide*
- *ServerIron ADX Security Guide*
- *ServerIron ADX Administration Guide*
- *ServerIron ADX Switch and Router Guide*
- *ServerIron ADX Firewall Load Balancing Guide*
- *ServerIron Hardware Installation Guide*
- *Ironware MIB Reference Manual*

Getting technical help or reporting errors

Brocade is committed to ensuring that your investment in our products remains cost-effective. If you need assistance, or find errors in the manuals, contact Brocade using one of the following options:

Web access

The Knowledge Portal (KP) contains the latest version of this guide and other user guides for the product. You can also report errors on the KP.

Log in to my.Brocade.com, click the **Product Documentation** tab, then click on the link to the Knowledge Portal (KP). Then click on **Cases > Create a New Ticket** to report an error. Make sure you specify the document title in the ticket description.

E-mail and telephone access

Go to <http://www.brocade.com/services-support/index.page> for the latest e-mail and telephone contact information.

In this chapter

ServerIron System Management

Setting up local user accounts

For each user account, you specify the user name. You can also specify:

- A password
- The privilege level, which can be one of the following:
 - Full access (super-user). This is the default.
 - Port-configuration access
 - Read-only access

To configure user accounts, you must add a user account for super-user access before you can add accounts for other access levels. You will need the super-user account to make further administrative changes.

You must be logged on with super-user access (privilege level 0, or with a valid Enable password for super-user access) to add user accounts or configure other access parameters.

To set up local user accounts, enter following commands.

```
ServerIronADX(config)# username greg-mcmillan nopassword
ServerIronADX(config)# username waldo privilege 5 password whereis
```

The first command adds a user account for a super-user with the user name "greg-mcmillan" and no password with privilege level super-user. This user has full access to all configuration and display features.

The second command adds a user account for user name "waldo", password "whereis", with privilege level read-only. Waldo can look for information but cannot make configuration changes.

Syntax: **[no] username** *<user-string>* **privilege** *<privilege-level>* **password** *<password-string>* | **nopassword**

The **privilege** *<privilege-level>* parameter specifies one of the following:

- 0 – Full access (super-user)
- 4 – Port-configuration access
- 5 – Read-only access

The default privilege level is 0. To assign full access to the user account, you can enter the command without **privilege 0**, as shown in the command example above.

The **password** | **nopassword** parameter indicates whether the user must enter a password. If you specify **password**, enter the string for the user's password.

NOTE

There is a default username "admin" and the password "brocade". For the security purpose, you may want to delete the default username. You will have to create at least one username in order to delete it. Otherwise, the default username will be automatically created after rebooted.

Displaying user information

To display user information, enter the following command.

```
ServerIronADX(config)# show users
Username          Password          Encrypt   Priv Status
Expire Time
=====
=====
admin             $1$T62..hul$hmRolcVlVwc.FCtXVD6h9/ enabled    0    enabled
Never
```

Configuring Telnet

The ServerIronADX supports up to five concurrent inbound Telnet and SSH sessions, one outbound Telnet session, and console access. Write access through Telnet and SSH is limited to one session only.

To access the CLI shell running Switch (S) code, Telnet or SSH to the assigned management IP address, assuming your client is on the same subnet of course.

```
ServerIronADX(config)# ip address 10.1.1.1 255.255.255.0
```

```
ServerIronADX(config)#show ip
Switch IP address: 10.1.1.1
Subnet mask: 255.255.255.0
Default router address: 10.1.1.2
Default IP MTU (Bytes): 1500
TFTP server address: None
Configuration filename: None
Image filename: None
```

If you are on a different subnet and running Switch code, configure an **ip default-gateway** *<ip-addr>*. This command also assists SNMP management.

If you are running Router (R) code, the management **ip address** must be set on a reachable system interface (physical or virtual). Use **ip route** 0.0.0.0 0.0.0.0 [*<next-hop-ip>*] to install a static route in R code.

Use **show who** or **show telnet** to display both Telnet and SSH user session information.

```
ServerIronADX# show who
Console connections:
    established
    you are connecting to this session
    1 seconds in idle
Telnet connections (inbound):
  1      closed
  2      closed
  3      closed
  4      closed
  5      closed
Telnet connection (outbound):
  6      closed
SSH connections:
  1      closed
  2      closed
  3      closed
  4      closed
  5      closed
```

Enabling Telnet authentication

To use local access control or a RADIUS server to authenticate telnet access to the ServerIron ADX, enter the following command.

```
ServerIronADX(config)# enable telnet authentication
```

Syntax: [no] enable telnet authentication

Enabling Telnet password

To assign a password for Telnet session access, enter the following command.

```
ServerIronADX(config)# enable telnet password secretsalso
```

Syntax: [no] enable telnet password <text>

The <text> parameter specifies the password and is up to 48 alphanumeric characters.

To close a Telnet session, enter **logout**.

Using a standard ACL to control Telnet access

You can apply an ACL to control Telnet access to the device.

The following commands configure ACL 10, then apply the ACL as the access list for Telnet access. The device will allow Telnet access to all IP addresses except those listed in ACL 10.

```
ServerIronADX(config)# access-list 10 deny host 209.157.22.32 log
ServerIronADX(config)# access-list 10 deny 209.157.23.0 0.0.0.255 log
ServerIronADX(config)# access-list 10 deny 209.157.24.0 0.0.0.255 log
ServerIronADX(config)# access-list 10 deny 209.157.25.0/24 log
ServerIronADX(config)# access-list 10 permit any
ServerIronADX(config)# telnet access-group 10
```

Syntax: [no] telnet access-group <num>

The <num> parameter specifies the number of a standard ACL and must be from 1 – 99.

Restricting Telnet management access

You can restrict Telnet management access to the Brocade device to the host whose IP address you specify. No other device except the one with the specified IP address can access the Brocade device's CLI through Telnet. You can use the command up to ten times for up to ten IP addresses.

If you want to restrict access from SNMP or the Web, use one or two of the following commands:

- `snmp-client` – restricts SNMP access (including IronView).
- `web client` – restricts web access.

If you want to restrict all management access, you can use the commands above and the **telnet client** command or you can use the following command: **all-client**.

To restrict Telnet access (which includes IronView) to the Brocade device to the host with IP address 209.157.22.26, enter the following command.

```
ServerIronADX(config)# telnet client 209.157.22.26
```

Syntax: `[no] telnet client <ip-addr>`

Changing the Telnet login timeout period

To change the login timeout period for Telnet sessions, enter the following command.

```
ServerIronADX(config)# telnet login-timeout 5
```

Syntax: `[no] telnet login-timeout <minutes>`

The *<minutes>* parameter specifies 1 – 10 minutes. The default is 2 minutes.

Enabling or disabling Telnet access

By default, Telnet access is enabled on the system.

To disable Telnet access to a ServerIron ADX, enter the following command.

```
ServerIronADX(config)# no telnet server
```

Syntax: `[no] telnet server`

Allowing Telnet access only to clients in a VLAN

You can allow Telnet access only to clients in a specific VLAN.

The following command configures the device to allow Telnet management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

```
ServerIronADX(config)# telnet server enable vlan 10
```

Syntax: `[no] telnet server enable vlan <vlan-id>`

Suppressing the rejection message

You can suppress the rejection message the device sends in response to a denied Telnet client.

If you enable suppression of the connection rejection message, a denied Telnet client does not receive a message from the device. Instead, the denied client simply does not gain access.

To suppress the connection rejection message sent by the device to a denied Telnet client, enter the following command.

```
ServerIronADX(config)# telnet server suppress-reject-message
```

Syntax: [no] telnet server suppress-reject-message

Defining Telnet timeout

By default, Telnet sessions do not time out (0 minutes).

To define how long a Telnet session can remain idle before it is timed out, enter the following command.

```
ServerIronADX(config)#telnet timeout 120
```

Syntax: [no] telnet timeout <minutes>

Configuring SSH

The ServerIron ADX supports up to five concurrent inbound Telnet and SSH sessions, one outbound Telnet session, and console access. Write access through Telnet and SSH is limited to one session only.

NOTE

SSH public key authentication supports only DSA keys. RSA keys are not supported.

Enabling or disabling SSH service

The SSH service is not enabled by default. The SSH server starts once you configure a host DSA public and private key pair for SSH.

```
ServerIronADX(config)# crypto key generate dsa
ServerIronADX(config)# write mem
```

Syntax: [no] crypto key generate dsa

The host DSA key pair is stored in the system-config file. Only the public key is readable. The host DSA key pair is used to negotiate a session key and encryption method with the SSH clients trying to connect to the ServerIron ADX.

The service is stopped once the keys are destroyed from the system-config file.

```
ServerIronADX(config)# crypto key zeroize dsa
ServerIronADX(config)# write mem
```

Syntax: crypto key zeroize dsa

There is no SSH client within the ServerIronADX to support outbound sessions initiated from within the ServerIronADX.

At a minimum, the following SSH clients are supported for inbound connections:

- F-Secure 5.3
- Secure Shell 3.2.3
- SecureCRT 4.0
- PuTTY 0.54
- Tera Term Pro 3.1.3
- OpenSSH_3.5p1

Creating a seed for generating a random number

To create a new seed for generating a random number that is used for generating the dynamically created server DSA key pair for SSH, enter the following command.

```
ServerIronADX(config)# crypto random-number-seed generate
```

Syntax: [no] crypto random-number-seed

Setting SSH authentication retries

To set the number of SSH authentication retries, enter the following command.

```
ServerIronADX(config)# ip ssh authentication-retries 5
```

Syntax: [no] ip ssh authentication-retries <number>

The <number> parameter can be from 1 to 5. The default is 3.

Setting the SSH key size

The size of the *host RSA* key that resides in the system-config file is always 1024 bits and cannot be changed.

To set the SSH key size, enter the following command.

```
ServerIronADX(config)# ip ssh key-size 896
```

Syntax: [no] ip ssh key-size <number>

The <number> parameter can be from 512 – 896 bits. The default is 768 bits.

Configuring SSH password authentication

By default, SSH password authentication is enabled.

After the SSH server on the Brocade device negotiates a session key and encryption method with the connecting client, user authentication takes place. Of the methods of user authentication available in SSH, Brocade's implementation of SSH supports password authentication only.

With password authentication, users are prompted for a password when they attempt to log into the device (unless empty password logins are not allowed; see **ip ssh permit-empty-passwd**). If there is no user account that matches the user name and password supplied by the user, the user is not granted access.

You can deactivate password authentication for SSH. However, since password authentication is the only user authentication method supported for SSH, this means that no user authentication is performed at all. Deactivating password authentication essentially disables the SSH server entirely.

To deactivate password authentication, enter the following command.

```
ServerIronADX(config)# ip ssh password-authentication no
```

Syntax: [no] ip ssh password-authentication no | yes

The **yes** option enables SSH password authentication.

Enabling empty password logins

By default, empty password logins are not allowed. This means that users with an SSH client are always prompted for a password when they log into the device. To gain access to the device, each user must have a user name and password. .

If you enable empty password logins, users are not prompted for a password when they log in. Any user with an SSH client can log in without being prompted for a password.

To enable empty password logins, enter the following command.

```
ServerIronADX(config)# ip ssh permit-empty-passwd yes
```

Syntax: [no] ip ssh permit-empty-passwd no | yes

The **yes** option enables SSH empty password login.

Changing the TCP port used for SSH

By default, SSH traffic occurs on TCP port 22.

To change the TCP port used for SSH, enter the following command.

```
ServerIronADX(config)# ip ssh port 2200
```

Syntax: [no] ip ssh port <number>

The <number> parameter specifies a valid TCP port number.

Note that if you change the default SSH port number, you must configure SSH clients to connect to the new port. Also, you should be careful not to assign SSH to a port that is used by another service. If you change the SSH port number, We recommend that you change it to a port number greater than 1024.

Loading a public key file

NOTE

SSH public key authentication supports only DSA keys. RSA keys are not supported.

To cause a public key file to be loaded onto the device, enter commands such as the following.

```
ServerIronADX(config)# ip ssh pub-key-file slot1 pkeys.txt
ServerIronADX(config)# ip ssh pub-key-file tftp 192.168.1.234 pkeys.txt
ServerIronADX(config)# ip ssh pub-key-file reload
ServerIronADX(config)# ip ssh pub-key-file flash-memory
ServerIronADX(config)# write memory
```

Syntax: [no] ip ssh pub-key-file slot1 | slot2 <filename>

Syntax: [no] ip ssh pub-key-file tftp <tftp-server-ip-addr> <filename>

Syntax: [no] ip ssh pub-key-file reload

Syntax: [no] ip ssh pub-key-file flash-memory

The **slot1 | slot2 <filename>** parameter causes a public key file called <filename> to be loaded from the Management IV module's PCMCIA flash card each time the device is booted.

The **tftp <tftp-server-ip-addr> <filename>** parameter causes a public key file called <filename> to be loaded from a TFTP server each time the Brocade device is booted.

The **reload** keyword reloads the public keys from the file on the TFTP server or PCMCIA flash card.

The **flash-memory** keyword makes the public keys in the active configuration part of the startup-config file.

Disabling or re-enabling RSA challenge-response authentication

RSA challenge-response authentication is enabled by default.

To disable RSA challenge-response authentication, enter the following command.

```
ServerIronADX(config)# ip ssh rsa-authentication no
```


Syntax: [no] ip ssh rsa-authentication yes | no

The **yes** option enables RSA challenge-response authentication.

Disabling or re-enabling Secure Copy

Secure Copy (SCP) is enabled by default.

To disable SCP, enter the following command.

```
ServerIronADX(config)# ip ssh scp disable
```

Syntax: [no] ip ssh scp disable | enable

NOTE

If you disable SSH, SCP is also disabled.

Using Secure Copy

Secure Copy (SCP) uses security built into SSH to transfer files between hosts on a network, providing a more secure file transfer method than Remote Copy (RCP) or FTP. SCP automatically uses the authentication methods, encryption algorithm, and data compression level configured for SSH. For example, if password authentication is enabled for SSH, the user is prompted for a user name and password before SCP allows a file to be transferred. No additional configuration is required for SCP on top of SSH.

You can use SCP to copy files on the device, including the startup-config and running-config files, to or from an SCP-enabled remote host.

SCP is enabled by default and can be disabled. To disable SCP, enter the following command.

```
ServerIronADX(config)# ip ssh scp disable
```

Syntax: [no] ip ssh scp disable | enable

If you disable SSH, SCP is also disabled.

The following are examples of using SCP to transfer files from and to a ServerIron ADX.

When using SCP, you enter the scp commands on the SCP-enabled client, rather than the console on the ServerIron ADX.

Certain SCP client options, including -p and -r, are ignored by the SCP server. If an option is ignored, the client is notified.

To copy a configuration file (c:\cfg\brocade.cfg) to the running-config file on a device at 192.168.1.50 and log in as user terry, enter the following command on the SCP-enabled client.

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:runConfig
```

If password authentication is enabled for SSH, the user is prompted for user terry's password before the file transfer takes place.

To copy the configuration file to the startup-config file.

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:startConfig
```

To copy the configuration file to a file called config1.cfg on the PCMCIA flash card in slot 1 on a Management IV module.

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:a:/config1.cfg
```

1 Configuring SSH

To copy the configuration file to a file called config1.cfg on the PCMCIA flash card in slot 2 on a Management IV module.

```
C:\> scp c:\cfg\brocade.cfg terry@192.168.1.50:b:/config1.cfg
```

To copy the running-config file on a ServerIron ADX to a file called c:\cfg\brcdhprun.cfg on the SCP-enabled client.

```
C:\> scp terry@192.168.1.50:runConfig c:\cfg\brcdhprun.cfg
```

To copy the startup-config file on a ServerIron ADX to a file called c:\cfg\brcdhstart.cfg on the SCP-enabled client.

```
C:\> scp terry@192.168.1.50:startConfig c:\cfg\brcdhstart.cfg
```

To copy a file called config1.cfg on the PCMCIA flash card in slot 1 on a Management IV module to the SCP-enabled client.

```
C:\> scp terry@192.168.1.50:a:/config1.cfg c:\cfg\config1.cfg
```

To copy a file called config2.cfg on the PCMCIA flash card in slot 1 on a Management IV module to the SCP-enabled client.

```
C:\> scp terry@192.168.1.50:b:/config2.cfg c:\cfg\config2.cfg
```

Defining the SSH timeout value

When the SSH server attempts to negotiate a session key and encryption method with a connecting client, it waits a maximum of 120 seconds for a response from the client. If there is no response from the client after 120 seconds, the SSH server disconnects.

To change the SSH timeout value, enter the following command.

```
ServerIronADX(config)# ip ssh timeout 60
```

Syntax: [no] ip ssh timeout <seconds>

The <seconds> parameter is from 1 to 120 seconds. The default is 120.

Using a standard ACL to control SSH access

You can apply an ACL to control SSH access to the device.

The following commands configure ACL 10, then apply the ACL as the access list for SSH access. The device will allow SSH access to all IP addresses except those listed in ACL 10.

```
ServerIronADX(config)# access-list 10 deny host 209.157.22.32 log
ServerIronADX(config)# access-list 10 deny 209.157.23.0 0.0.0.255 log
ServerIronADX(config)# access-list 10 deny 209.157.24.0 0.0.0.255 log
ServerIronADX(config)# access-list 10 deny 209.157.25.0/24 log
ServerIronADX(config)# access-list 10 permit any
ServerIronADX(config)# ssh access-group 10
```

Syntax: [no] ssh access-group <num>

The <num> parameter specifies the number of a standard ACL and must be from 1 – 99.

Displaying SSH information

To display detailed SSH information, enter the following command.

```
ServerIronADX(config)# show ip ssh
Connection      Version      Encryption    Username
1               SSH-2       3des-cbc      admin
```

Syntax: show ip ssh

Displaying currently loaded public keys

To display the currently loaded public keys, enter the following command.

```
ServerIronADX# show ip client-public-key
1024 65537 162566050678380006149460550286514061230306797782065166110686648548574
94957339232259963157379681924847634614532742178652767231995746941441604714682680
00644536790333042029124905690771828865418 39656556769025432881477252978135927821
67540629478392662275128774861815448523997023618173312328476660721888873946758201
user@csp_client

1024 35 152676199889856769693556155614587291553826312328095300428421494164360924
76207475545234679268443233762295312979418833525975695775705101805212541008074877
26586119857422702897004112168852145074087969840642408451742714558592361693705908
74837875599405503479603024287131312793895007927438074972787423695977635251943 ro
ot@unix_machine
```

There are 2 authorized client public keys configured

Syntax: show ip client-public-key

Managing System Functions

This section contains information on Managing the System Functions

Terminating the active CLI session

You can terminate the specified active CLI session and reset the configuration token. Once you know the session ID of a Telnet connection (use the **show who** command), you can terminate it with the **kill** command. If the terminated session was a console, the console is sent back into User EXEC mode. If the terminated CLI session was a Telnet or SSH session, the connection is closed.

```
ServerIronADX# kill telnet 1
```

Syntax: kill {console | telnet <session-id> | ssh <session-id>}

Performing a lookup on a domain

To perform a lookup on a specified domain, enter the following command.

```
ServerIronADX# whois boole.com
```

Syntax: whois <host-ip-addr> | <domain>

The <host-ip-addr> parameter is a valid IP address and <domain> is a valid domain name. A DNS gateway must be defined in order to use this command.

Verifying connectivity

The **ping** command verifies connectivity to a device. The command performs an ICMP echo test. An ICMP Request goes to the target host, and the host sends back an ICMP Reply packet. You can send a test packet to a host's IP address or host name.

The ServerIronADX can **ping** using arbitrary source IP addresses (Src-IPs) belonging to the device. The *<source-ip-addr>* was the management IP of the switch by default. You have the flexibility to use any *<source-ip-addr>* belonging to the device.

To verify connectivity to a device, enter the ping command such as the following.

```
ServerIronADX> ping 192.22.2.33
```

Syntax: **ping** *<dest-ip-addr>* | *<hostname>* [*<source-ip-addr>*] [**count** *<num>*] [**timeout** *<msec>*] [**ttl** *<num>*] [**size** *<byte>*] [**quiet**] [**numeric**] [**no-fragment**] [**verify**] [**data** *<1-to-4 byte hex>*] [**brief**]

The *<hostname>* parameter can be used only if you have already enabled the Domain Name Server (DNS) resolver feature on the device from which you are sending the ping. See **ip dns domain-name** and **ip dns server-address**.

The *<dest-ip-addr>* parameter specifies the IP address to be used as the destination of the ping packets.

The *<source-ip-addr>* parameter specifies the IP address to be used as the source (origin) of the ping packets.

The **count** *<num>* parameter specifies the number of ping packets the device sends. You can specify from 1 – 4294967296. The default is 1.

The **timeout** *<msec>* parameter specifies the number of milliseconds the Brocade device waits for a reply from the pinged device. You can specify a timeout from 1 – 4294967296 milliseconds. The default is 5000 (5 seconds).

The **ttl** *<num>* parameter specifies the maximum number of hops. You can specify a TTL from 1 – 255. The default is 64.

The **size** *<byte>* parameter specifies the size of the ICMP data portion of the packet. This is the payload and does not include the header. You can specify from 0 – 4000. The default is 16.

The **no-fragment** option turns on the “do not fragment” bit in the IP header of the ping packet. This option is disabled by default.

The **quiet** option hides informational messages such as a summary of the ping parameters sent to the device and instead only displays messages indicating the success or failure of the ping. This option is disabled by default.

The **verify** option ensures the data in the echo packet (the reply packet) is the same as the data in the echo request (the ping). By default the device does not verify the data.

The **data** *<1 – 4 byte hex>* parameter specifies a specific data pattern for the payload instead of the default data pattern, “abcd”, in the packet's data payload. The pattern repeats itself throughout the ICMP message (payload) portion of the packet. For numeric parameter values, the CLI does not check that the value you enter is within the allowed range. Instead, if you do exceed the range for a numeric value, the software rounds the value to the nearest valid value.

The **brief** parameter causes ping test characters to be displayed. The following ping test characters are supported:

! —Indicates that a reply was received.

- Indicates that the network server timed out while waiting for a reply.
- ⌵—Indicates that a destination unreachable error PDU was received.
- ⌵—Indicates that the user interrupted ping.

If you address the ping to the IP broadcast address, the device lists the first four responses to the ping.

Tracing the IP path to a host

The **traceroute** command enables you to trace the IP path to a host. It displays a list of all the intervening router hops the trace-route request traversed to reach the host. In addition, if there are multiple equal-cost routes to the destination, the Brocade device displays up to three responses by default.

To perform a traceroute, enter a command such as the following.

```
ServerIronADX> traceroute 192.33.4.7 minttl 5 maxttl 5 timeout 5
```

Syntax: **traceroute** *<host-ip-addr>* [**maxttl** *<value>*] [**minttl** *<value>*] [**numeric**] [**timeout** *<value>*] [**source-ip** *<ip addr>*]

The **minttl** parameter specifies the minimum TTL (hops) value. Possible values are 1 – 255. The default is 1 second.

The **maxttl** parameter specifies the maximum TTL (hops) value. Possible values are 1 – 255. The default is 30 seconds.

The **timeout** value can be from 1 – 120. The default is 2 seconds.

The **numeric** option changes the display to list the devices by their IP addresses instead of their names.

The **source-ip** *<ip addr>* parameter specifies an IP address to be used as the origin for the traceroute.

To halt an initiated trace, enter the following command.

```
ServerIronADX> stop-traceroute
```

Syntax: **stop-traceroute**

Initiating a system reset

Use the **reload** command to initiate a system reset. You will be prompted to save all configuration changes made since the last reset or start of the ServerIron ADX to the startup configuration file.

Although the dynamic configuration feature allows many parameter changes to take effect immediately without a system reset, other parameters do require a system reset. To place these parameters into effect, you must save the configuration changes to the configuration file, then reload the system. The management interfaces provide an option to immediately reset the system. Alternatively, you can use the scheduled system reload feature to configure the system to reload its flash code at a specific time (based on the system time counter or SNTP time) or after a specific amount of time has passed.

To initiate a system reset, enter the following command.

```
ServerIronADX# reload
```

Syntax: `reload [after <dd:hh:mm>] | [at <hh:mm:ss> <mm-dd-yy>] | [cancel] [primary | secondary]`

The `after <dd:hh:mm>` parameter reloads after the specified amount of time has passed.

The `at <hh:mm:ss> <mm-dd-yy>` parameter reloads at exactly the specified time.

The `cancel` option negates the scheduled reload.

The `primary | secondary` option specifies whether the reload is to occur from the primary code flash module or the secondary code flash module. The default is primary.

The **reload** command must be typed in its entirety.

Logging into a BP

To log into a Barrel Processor (BP) on the Application Switching Module card, enter the following **rconsole** command.

```
ServerIronADX# rconsole 1 1
ServerIronADX1/1#

asm                show all application switch module commands
rcon-exit          Exit rconsole
rconsole-exit      Exit rconsole
show               Display system information
write              Write running configuration to terminal
ServerIronADX1/1# rconsole-exit
```

NOTE

A BP is the Applications traffic switching processor.

The example moves the CLI session from the management processor (MP) to BP 1 on the Application Switching Module in slot 1. Notice the end of the command prompt changes to indicate the ASM slot number and BP number.

Syntax: `rconsole <asm-slot-number> <bp-number>`

The `<asm-slot-number>` variable specifies the chassis slot containing the module (see **show module**).

The chassis slots specified in the `<asm-slot-number>` variable are numbered 1 - 2 from top to bottom in a ServerIron ADX 4000 chassis.

The chassis slots specified in the `<asm-slot-number>` variable are numbered 1 - 4 from top to bottom in a ServerIron ADX 8000 chassis.

The slot specified in the `<asm-slot-number>` variable is always 1 in a ServerIron ADX 1000.

The `<bp-number>` parameter specifies the BP (numbered from 1 – 8 maximum).

Use the **rconsole-exit** command to return to the MP.

Timing out idle serial management sessions

You can time out idle serial management sessions. By default, a device does not time out serial CLI sessions. A serial session remains open indefinitely until you close it.

NOTE

If a session times out, the device does not close the connection. Instead, the CLI changes to the User EXEC mode (for example: `ServerIronADX>`).

To time out idle serial management sessions, enter the following command.

```
ServerIronADX(config)#console timeout 20
```

Syntax: `[no] console timeout <num>`

The `<num>` parameter specifies the number of minutes, from 0 – 240, that the serial CLI session can remain idle before it times out. The default is 0 (sessions never time out).

Configuring a ServerIron ADX to broadcast a session delete message

To configure the ServerIron ADX to broadcast a session delete message to all of its BPs when it deletes a server's session table entry pair, enter the following command.

```
ServerIronADX(config)#server udp-bc-client-session-del
```

Syntax: `[no] server udp-bc-client-session-del`

This command applies only to configurations where a client is connected to a router that is not the ServerIron ADX's default gateway, and which is handled by a BP that does not also handle the ServerIron ADX's default gateway.

Assigning a name to the ServerIron ADX

You can assign a name to the device, by entering a command such as the following.

```
ServerIronADX(config)# hostname chassis  
ServerIronADX(config)#
```

Syntax: `[no] hostname <text>`

The `<text>` parameter can be up to 32 alphanumeric characters.

Assigning an administrative ID

You can assign an administrative ID to the device, by entering a command such as the following.

```
ServerIronADX(config)# chassis name routernyc
```

Syntax: `[no] chassis name <text>`

The `<text>` parameter is up to 32 alphanumeric characters.

This command does not change the CLI prompt. To change the CLI prompt, use the **hostname** command.

Disabling or re-enabling password encryption

Password encryption is enabled by default. When encryption is enabled, users cannot learn the device's passwords by viewing the configuration file.

Password encryption does not encrypt the password in Telnet packets sent to the device. This feature applies only to the configuration file.

To disable password encryption, enter the following command.

```
ServerIronADX(config)# no service password-encryption
```

Syntax: [no] service password-encryption

Understanding dynamic configuration

In most cases, dynamic configuration enables you to make configuration changes without rebooting the system. Most Layer 2 configuration changes are dynamic. All Layer 4-7 configuration changes are dynamic.

If a command requires a **reload** to be effective, the device will display this information after the command is entered. Where reload is needed use the **system-max** command.

Disabling or re-enabling the page-display mode

The page-display mode displays the file one page at a time and prompts you to continue or cancel the display. When page-display mode is disabled, if you display or save the configuration file, the CLI displays the entire file without interruption.

By default, the page-display mode is enabled. When the ServerIron ADX prints text, one "page" (window-full) of the file is displayed. The following line provides you with options to continue the display or to cancel with Ctrl-c.

```
--More--, next page: Space/Return key, quit: Control-c
```

To disable the page-display mode, enter the following command.

```
ServerIronADX# skip-page-display  
Disable page display mode
```

To enable the page-display mode, enter the following command.

```
ServerIronADX# page-display  
Enable page display mode
```

Syntax: skip-page-display

Syntax: page-display

Disabling or re-enabling the stop page display characteristic

You can remove the stop page display characteristic for the **write terminal** command.

For example, by default, when a user enters the command write terminal the full configuration will generally involve more than a single page display. You are prompted to enter the return key to view the next page of information. When this command is enabled, this page-by-page prompting will be removed and the entire display will roll on the screen until the end is reached.

To remove the stop page display characteristic for the **write terminal** command, enter the following command.

```
ServerIronADX(config)# enable skip-page-display
```

To re-enable the stop page display characteristic, enter **no enable skip-page-display**.

Syntax: [no] enable skip-page-display

Configuring a message for display at the Privileged EXEC level

You can configure the ServerIron ADX to display a message when a user enters the Privileged EXEC CLI level.

A delimiting character is established on the first line of the **banner exec** command. You begin and end the message with this delimiting character. It can be any character except " (double-quotation mark) and cannot appear in the banner text. The banner text can be up to 2047 characters long and can consist of multiple lines.

To configure the ServerIronADX to display a message when a user enters the Privileged EXEC CLI level, enter the following command.

```
ServerIronADX(config)# banner exec $ (Press Return)
Enter TEXT message, End with the character '$'.
You are entering Privileged EXEC level
Don't foul anything up! $
```

In this example, the delimiting character is \$ (dollar sign). The text in between the dollar signs is the contents of the banner.

To remove the banner, enter **no banner exec**.

Syntax: **[no] banner exec** *<delimiting-character>*

The *<delimiting-character>* parameter can be any character except " (double-quotation mark)

Configuring a message for display on a Console

You can configure the ServerIron ADX to display a message on the Console when a user establishes a Telnet session. This message indicates where the user is connecting from and displays a configurable text message.

To configure a message on the Console, enter the following.

```
ServerIronADX(config)# banner incoming $ (Press Return)
Enter TEXT message, End with the character '$'.
Incoming Telnet Session!! $
```

When a user connects to the CLI using Telnet, the following message appears on the Console.

```
Telnet from 209.157.22.63
Incoming Telnet Session!!
```

Syntax: **[no] banner incoming** *<delimiting-character>*

Configuring a message for display on a terminal

You can configure the ServerIronADX to display a message on a user's terminal when he or she establishes a Telnet CLI session.

To display the message "Welcome to ServerIron ADX!" when a Telnet CLI session is established, enter the following.

```
ServerIronADX(config)# banner motd $ (Press Return)
Enter TEXT message, End with the character '$'.
Welcome to ServerIron ADX! $
```

When you access the Web Management Interface, the banner is displayed on the login panel.

Syntax: `[no] banner <delimiting-character> | [motd <delimiting-character>]`

NOTE

The **banner** *<delimiting-character>* command is equivalent to the **banner motd** *<delimiting-character>* command.

Configuring TFTP

All Brocade devices allow you to use Trivial File Transfer Protocol (TFTP) to copy files to and from the flash memory modules on the management module. You can use TFTP to perform the following operations:

- Upgrade boot or flash code.
- Archive boot or flash code or a configuration file on a TFTP server.
- Load the system using flash code and a configuration file stored on a TFTP server. (This occurs as part of the BootP or DHCP process.)

NOTE

Certain boot upgrades may require you to install new firmware. Contact your reseller or Brocade Communications Systems Inc. for information.

Using the management port

All ServerIron ADX platforms have a 10/100/1000 Ethernet port that is designed to be used for managing the device. This port allows you to provide management access to a ServerIron ADX on a separate and more secure network that the one where general network traffic is being passed. This access is provided through an RJ-45 connector on the front panel of the ServerIron ADX 1000 platforms or on the management module for ServerIron ADX chassis products. [Figure 1](#) and [Figure 2](#) show the locations of the management ports on the ServerIron ADX 1000 and chassis devices.

FIGURE 1 Location of management port on ServerIron ADX 1000

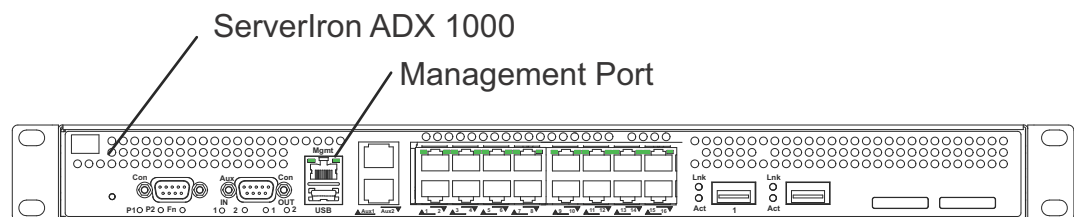
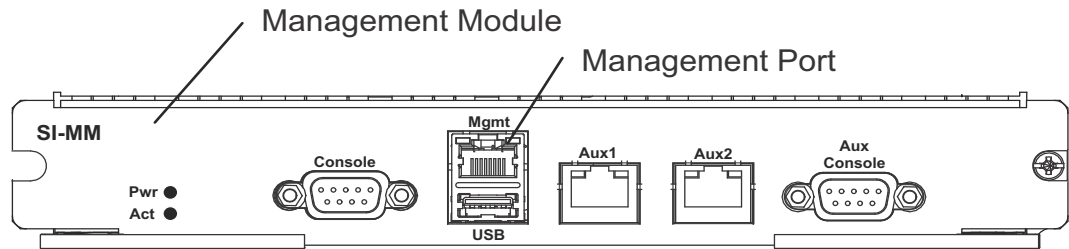


FIGURE 2 Location of management port on ServerIron ADX chassis devices



The ServerIron ADX management port functions as described in the following:

- The management port supports IPv4 operations and ARP packets. It does not support IPv6 operations.
- The management port allows you to configure and manage the ServerIron ADX only. As a result, this port has the same limited functionality as an IP host port.
- You cannot enable and run routing protocols on the management port.
- The management port supports static routes and directly connected routes, which are installed in the management module's routing table. However, these routes are not installed in the interface module's routing table. Therefore, the interface modules are not aware of the management port's static or directly connected routes.
- If you configure the redistribution of directly connected or static routes for a particular routing protocol, the routing protocol will redistribute directly connected or static routes associated with the interface module ports but not those associated with the management port.
- On a ServerIron ADX, the management port supports multiple static routes. With switch code, there is a restriction of 32 routes and these must be configured under the management interface configuration. With router code installed, you can configure static routes pointing to the management port at the global configuration level.
- You cannot configure a default route (0.0.0.0/0) that points to the management port.

To display configuration information and statistics about the management port, you can enter the `show interface management 1` command at any CLI level.

Configuring the management port

You can configure the ServerIron ADX management port for the following:

- To enable or disable the port
- An IP address
- An IP Route pointing over the management port (this is available with switch code only)

Enabling and disabling the management port

The interface port is enabled by default. It can be disabled using the **disable** command under the management interface configuration mode. Once disabled, it can be enabled using the **enable** command. The following example disables a management port on a ServerIron ADX.

```
ServerIronADX# configure terminal
ServerIronADX(config)# interface management 1
ServerIronADX(config-if-mgmt-1)# enable
```

Syntax: interface management 1

Syntax: enable | disable

Configuring an IP address on a management port

The management port can be configured with a distinct IP address that is different than an other IP address configured on the ServerIron ADX. This is true whether you are running switch or router code.

You can configure an IP address for the management port as shown in the following.

```
ServerIronADX# configure terminal
ServerIronADX(config)# interface management 1
ServerIronADX(config-if-mgmt-1)# ip address 10.10.10.1 255.255.255.0
```

Syntax: [no] ip address <IPaddress> <IPmask>

The <IPaddress> and <IPmask> variables specify the IP address that you want to assign to the management port.

NOTE

The IP subnet configured on the management port must be a distinct IP subnet and it should not overlap with the IP subnet configured on the global management IP on switch code or the IP subnet configured on the physical or virtual interfaces on router code. In addition, there should not be an IP subnet overlap between the management port and the IPs configured for any of the following:

- SLB/IP NAT/FWLB related features: source-ip, source-standby-ip, source-nat-ip, static NAT IP, dynamic NAT IP, virtual server ip, real server ip and virtual-ip (configured under server fw-group for FWLB).

The management port's IP address cannot be the same as the OS mode IP address that is configured in monitor mode using the **ip address addr** command and the subnet overlap is allowed in this situation.

The following example explains how to access monitor mode and OS mode.

To access the **monitor mode**, enter the following command.

```
ServerIronADX# reload
Are you sure? (enter 'y' or 'n'): y
Running Config data has been changed.
Do you want to continue the reload without saving the running config? (enter 'y'
or 'n'): y
Halt and reboot ServerIron Boot Code Version 12.0.0
Enter 'a' to stop at memory test
Enter 'b' to stop at boot monitor
***** Interrupted by entering 'b' *****
```

To access the **OS mode**, enter the following command.

```
ServerIronADX4000-vbp# -(Entering control Key plus Key 'Y' and the key 'M' on
the console will direct you to OS mode)
OS>
OS> sh remote
IP address : 10.18.26.10
subnet mask : 255.255.255.0
default gateway : 10.18.26.1
OS> exit -(This command will redirect you back to the Application
console)
ServerIronADX4000#
```

In addition, the management port should not be connected to the same vlan that has the data ports connecting from the ADX.

Configuring an IP route over the management port

You can configure up to 32 static routes over the management port. On switch code, in order to configure an IP static route on the management port with a destination address of 192.0.0.0 255.0.0.0 and a next-hop router IP address of 195.1.1.1 , enter the following commands.

```
ServerIronADX# configure terminal
ServerIronADX(config)# interface management 1
ServerIronADX(config-if-mgmt-1)# ip route 192.0.0.0 255.0.0.0 195.1.1.1
```

Syntax: `ip route <dest-ip-addr> <dest-mask>
<next-hop-ip-addr>`

The `<dest-ip-addr>` is the route's destination. The `<dest-mask>` is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/24.

The `<next-hop-ip-addr>` is the IP address of the next-hop router (gateway) for the route.

On router code, in order to configure an IP static route on the management port with a destination address of 192.0.0.0 255.0.0.0 and a next-hop router IP address of 195.1.1.1 , enter the following commands.

```
ServerIronADX# configure terminal
ServerIronADX(config)# ip route 192.0.0.0 255.0.0.0 195.1.1.1
```

Syntax: `ip route <dest-ip-addr> <dest-mask>
<next-hop-ip-addr> [distance <num>]`

The `<dest-ip-addr>` is the route's destination. The `<dest-mask>` is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/24.

The `<next-hop-ip-addr>` is the IP address of the next-hop router (gateway) for the route.

The **distance** `<num>` parameter specifies the administrative distance of the route. When comparing otherwise equal routes to a destination, the ServerIron ADX prefers lower administrative distances over higher ones, so make sure you use a low value for your default route. The default is 1.

Displaying management port information

You can obtain information about the management port using the following commands:

- show ip mgmt-route
- show ip route
- show interfaces brief
- show interfaces management 1
- show statistics management 1

1 Using the USB port and USB flash drive

The **show ip mgmt-route** command is shown in the following and the other commands are described in the *ServerIron ADX Switch and Router Guide*.

Displaying management port route information

On switch code you can display management port route information as shown in the following.

```
ServerIronADX(config-if-mgmt-1)# show ip mgmt-route
Total number of IP routes: 1
```

	Destination	NetMask	Gateway	Port	Cost	Type
1	10.10.10.0	255.255.255.0	0.0.0.0	mgmt1	1	D
2	192.0.0.0	255.0.0.0	10.10.10.254	mgmt1	1	S

Syntax: show ip mgmt-route

On router code, you display management port route information as shown in the following.

```
ServerIronADX# show ip route
Total number of IP routes: 2
Start index: 1 D:Connected R:RIP S:Static O:OSPF *:Candidate default
```

	Destination	NetMask	Gateway	Port	Cost	Type
1	10.10.10.0	255.255.255.0	0.0.0.0	mgmt1	1	D(N)
2	192.0.0.0	255.255.255.0	10.10.10.254	mgmt1	1	S(N)

Syntax: show ip route

Using the USB port and USB flash drive

All ServerIron ADX models are equipped with an internal and an external USB port. The internal (**usb0**) port is a USB drive with 4 GB of memory. The external (**usb1**) port points to a USB connector on the ServerIron ADX switch where you can connect an external USB flash drive.

NOTE

The external USB port (**usb1**) does not support USB hard drives.

The following sections describe procedures for:

- Creating directories on USB drives.
- Removing directories on USB drives.
- Displaying contents of USB drive directories.
- Copying files between ServerIron ADX code flash and USB drives.
- Copying files between an tftp server, ServerIron ADX code flash, and USB drives.
- Renaming files on USB drives.
- Deleting files from USB drives and ServerIron ADX code flash.
- Formatting USB drives.
- Testing USB drives.

Creating directories on USB drives

Using the **mkdir** and **md** commands You can create up to two directory levels on an internal USB drive (**usb0**) or USB drive attached to the external USB port (**usb1**) and can copy files to or remove files from these directories. As an example, a first-level directory structure on **usb0** would be **usb0/dir1**. An example of a second-level directory structure would be **usb1/dir1/dir2**. Following is a list of rules for creating directories:

- You cannot create a directory under **usb0/sys**, as this contains system files. For example, the system creates the “**dpscript**” directory under **usb0/sys/** to store scripts for the scripting feature.
- In a dual MP (management processor) environment, files in **usb0/sys/dpscript** will be automatically synchronized between MPs.
- You can use both “/” or “\” as directory delimiters.
- Although you can copy files to and from directories, you cannot copy entire directories.
- You can delete or rename files in directories.

Following are examples of using the **mkdir** or **md** commands to create directories:

- Use the following commands to create a directory called *dir1* on the **usb1** drive.

```
ServerIronADX# mkdir usb1/dir1
ServerIronADX# md usb1/dir1
```

- Use the following commands to create a first-level directory called “*abcd*” and a second-level directory called “*dir2*” on the **usb0** drive.

```
ServerIronADX# mkdir usb0/abcd/dir2
ServerIronADX# md usb0/abcd/dir2
```

Syntax: **mkdir** <dir-full-path>

or

Syntax: **md** <dir-full-path>

The **mkdir** or **md** command creates a directory.

The <dir-full-path> parameter specifies the internal USB drive (**usb0**) or the external USB port (**usb1**) on the ServerIron ADX connected to a USB drive and full path of the directory. For example, the full path for a second-level directory “*dir2*” on **usb0** could be **usb0/dir1/dir2**. You must use the full path when creating a directory using “/” or “\” as directory delimiters.

Removing directories on USB drives

Using the **rmdir** command, you can remove empty directories only, and you must specify the full path for these directories.

Following are examples of using this command:

- Use the following command to remove the empty first-level directory “*dir1*” from the **usb1** drive.

```
ServerIronADX# rmdir usb1/dir1
```

- Use the following command to remove the empty second-level directory “*dir2*” from the **usb0** drive.

```
ServerIronADX# rmdir usb0/dir1/dir2
```

1 Using the USB port and USB flash drive

- Use the following command to remove the first-level directory *dirA* from the USB drive connected to the external USB port (**usb1**).

```
ServerIronADX# rd usb1/dirA
```

Syntax: **rmdir** *<dir-full-path>*

or

Syntax: **rd** *<dir-full-path>*

The **rmdir** or **rd** command removes a directory.

The *<dir-full-path>* parameter specifies the internal USB drive (**usb0**) or the external USB port (**usb1**) on the ServerIron ADX connected to a USB drive and full path of the directory. For example, the full path for a second-level directory "dir2" on **usb0** could be **usb0/dir1/dir2**. You must use the full path when creating a directory using "/" or "\" as directory delimiters.

Displaying contents of USB drive directories

Using the **dir** command, you can display the contents of a the root level of code flash, **usb0**, or **usb1** or display contents of specific directories created in **usb0** and **usb1**.

Following are examples of using this command:

- Use the **dir** command without parameters, as in the following example, displays the contents of the root level of code flash, **usb0**, and **usb1**.

```
ServerIronADX# dir
```

- Use the following command to display the contents of the root level of **usb0**. You can use **usb0** (internal USB drive) or **usb1** (external USB port on the ServerIron ADX where a USB drive can connect).

```
ServerIronADX# dir usb0
```

- Use the following command to display the contents of "abcd" in the path "usb1/abcd".

```
ServerIronADX# dir usb1/abcd
```

- Use the following command to display the contents of "dir2" in the path "usb0/dir1/dir2".

```
ServerIronADX# dir usb0/dir1/dir2
```

Syntax: **dir** *<dir-full-path>*

The **dir** command displays the contents of the specified directory.

The *<dir-full-path>* parameter specifies the internal USB drive (**usb0**) or the external USB port (**usb1**) on the ServerIron ADX connected to a USB drive and full path of the directory. For example, the full path for a second-level directory "dir2" on **usb0** could be **usb0/dir1/dir2**. You must use the full path when creating a directory using "/" or "\" as directory delimiters.

Copying files between flash and USB drives

Using the **copy** command, you can copy a file from code flash to a USB drive or from a USB drive (internal or external) to code flash.

The following command copies the primary image file from flash to "my_file" in a second-level directory (dir1) of **usb1**.


```
copy flash usb1 primary /abc/dir1/my_file
```

Syntax: **copy flash** *<destination-usb>* *primary/secondary* *<dest-full-path>*

The **copy** command copies the specified file.

Flash specifies that the file resides in ServerIron ADX code flash.

The *<destination-usb>* variable specifies the USB drive that the file will be copied to. The value can be either **usb0** (the internal USB drive) or **usb1** (a USB drive attached to the USB port on the ServerIron ADX).

Use *primary* or *secondary* when copying from a USB drive to flash. Copying to flash is restricted and allows image file names "primary" and "secondary". All files are located at the root level in code flash.

The *<dest-full-path>* variable specifies the full directory path and name of the file that you are copying to on the USB drive.

Following are examples of copying files from a USB drive to code flash:

- Use the following command to copy the file named "asm12000.bin" from the root-level of a drive attached to the USB port (**usb1**) on the ServerIron ADX to the secondary code file in code flash on the ServerIron ADX switch.

```
ServerIronADX# copy usb1 flash asm12000.bin secondary
```

- Use the following command to copy the file named "filexxx.bin" from the second-level directory "dir2" on a USB drive attached to the USB port (**usb1**) on the ServerIron ADX to the secondary code file in code flash on the ServerIron ADX.

```
ServerIronADX# copy usb1 flash /dir1/dir2/filexxx.bin secondary
```

Syntax: **copy** *<source-usb>* **flash** *<src-full-path>* *primary/secondary*

The **copy** command copies the specified file.

The *<source-usb>* variable specifies the USB drive that the file will be copied from. The value can be either **usb0** (the internal USB drive) or **usb1** (a USB drive attached to the USB port on the ServerIron ADX).

Flash specifies that the file resides on ServerIron ADX code flash.

The *<src-full-path>* variable specifies the full directory path and name of the file that you want to copy from the USB drive.

Use *primary* or *secondary* when copying from a USB drive to flash. Copying to flash is restricted and allows image file names "primary" and "secondary". All files are located at the root level in code flash.

Copying files between USB drives

Following are examples of using the **copy** command to copy files from a USB drive to another USB drive and copying files on a USB drive to another file on the same USB drive.

- Use the following command to copy the file named "asm12000.bin" on the root level of the Internal USB drive (**usb0**) to a file of the same name on the root level of a USB drive attached to the USB port (**usb1**) on a ServerIron ADX switch.

```
ServerIronADX# copy usb0 usb1 asm12000.bin asm12000.bin
```

1 Using the USB port and USB flash drive

- Use the following command to copy the file named "filexxx.bin" on the root level of the internal USB drive (usb0) to a file of the same name on the first-level directory "dir1" of a USB drive attached to the external USB port (usb1) on a ServerIron ADX switch.

```
ServerIronADX# copy usb0 usb1 filexxx.bin /dir1/filexxx.bin
```

- Use the following command to copy the file named "filexxx.bin" on the second-level directory of a USB drive attached to the USB port (usb1) on a ServerIron ADX switch to a file named "myfile.bin" on the first-level directory of the same USB drive.

```
ServerIronADX# copy usb1 usb1 /dir1/dir2/filexxx.bin /dir1/myfile.bin
```

Syntax: `copy <source-usb> <destination-usb> <src-full-path> <dest-full-path>`

The **copy** command copies the specified file.

The *<source-usb>* variable specifies the USB drive that the file will be copied from. The value can be either **usb0** (the internal USB drive) or **usb1** (a USB drive attached to the USB port on the ServerIron ADX).

The *<destination-usb>* variable specifies the USB drive that the file will be copied to. The value can be either **usb0** (the internal USB drive) or **usb1** (a USB drive attached to the USB port on the ServerIron ADX).

The *<src-full-path>* variable specifies the full directory path and name of the file that you want to copy from the USB drive.

The *<dest-full-path>* variable specifies the full directory path and name of the file that you are copying to on the USB drive.

Copying files between an tftp server, flash, and USB drives

Use the **copy** command to copy files between a Trivial File Transfer Protocol (**tftp**) server, a USB drive (**usb0** or **usb1**), and code flash (**flash**). You must obtain the IP address of the tftp server to use the copy command for this situations,

Example 1: Use the following command to copy the file "filexxx.bin" from an tftp server with address 1.1.1.1 to the "primary" code image in flash.

```
ServerIronADX# copy tftp flash 1.1.1.1 filexxx.bin primary
```

Syntax: `copy tftp flash <tftp-server-ip-addr> <src-file> primary/secondary`

The **copy** command copies the specified file.

tftp specifies that the file resides on the tftp server.

Flash specifies that the file resides on ServerIron ADX code flash.

The *<tftp-server-ip-addr>* parameter is the IP address of the tftp server. Note that this can be an IPV6 address (X:X::X:X).

The *<src-file>* parameter specifies the source file name.

Copying to flash is restricted and allows image file names "primary" and "secondary". All files are located at the root level in code flash.

Example 2: Use the following command to copy the "primary" code image from flash to a file named "file_abc" on an tftp server with address 1.1.1.1.

```
ServerIronADX# copy flash tftp 1.1.1.1 file_abc primary
```

Syntax: `copy flash tftp <tftp-server-ip-addr> <dest-file> primary/secondary`

The **copy** command copies the specified file.

Flash specifies that the file resides on ServerIron ADX code flash.

tftp specifies that the file resides on the tftp server.

The `<tftp-server-ip-addr>` parameter is the IP address of the tftp server. Note that this can be an IPV6 address (X:X::X:X).

The `<dest-file>` parameter specifies the destination file name.

The "primary" or "secondary" code image in flash. All files are located at the root level in code flash.

Example 3: Use the following command to copy the "secondary" code image from flash to the "primary" code image in flash.

```
ServerIronADX# copy flash flash primary
```

Syntax: `copy flash flash primary/secondary`

The **copy** command copies the specified file.

Flash specifies that the file resides on ServerIron ADX code flash.

Flash allows image files "primary" and "secondary". All files are located at the root level in code flash. Note that specifying "primary" copies the secondary to primary. Specifying "secondary" copies the primary to secondary.

Example 4: Use the following command to copy a file named "myfile.bin" on a tftp server with address 1.1.1.1 to a file named "myfile.bin" in the second-level directory "dir2" on internal USB drive (**usb0**).

```
ServerIronADX# copy tftp usb0 1.1.1.1 myfile.bin /dir1/dir2/myfile.bin
```

Syntax: `copy tftp <destination-usb> <tftp-server-ip-addr> <src-file> <dest-full-path>`

The **copy** command copies the specified file.

tftp specifies that the file resides on the tftp server.

The `<destination-usb>` variable specifies the USB drive that the file will be copied to. The value can be either **usb0** (the internal USB drive) or **usb1** (a USB drive attached to the USB port on the ServerIron ADX).

The `<tftp-server-ip-addr>` parameter is the IP address of the tftp server. Note that this can be an IPV6 address (X:X::X:X).

The `<src-file>` parameter specifies the source file name.

The `<dest-full-path>` variable specifies the full directory path and name of the file that you are copying to on the USB drive.

Example 5: Use the following command to copy a file named "asm12000.bin" in the first-level directory "abc" of a USB drive attached to the USB port (**usb1**) on the ServerIron ADX to a file named "myfile.bin" on a tftp server with address 1.1.1.1.

```
ServerIronADX# copy usb1 tftp /abc/asm12000.bin 1.1.1.1 myfile.bin
```

Syntax: `copy <source-usb> tftp <src-full-path> <tftp-server-ip-addr> <dest-file>`

The **copy** command copies the specified file.

1 Using the USB port and USB flash drive

The `<source-usb>` variable specifies the USB drive that the file will be copied from. The value can be either **usb0** (the internal USB drive) or **usb1** (a USB drive attached to the USB port on the ServerIron ADX).

tftp specifies that the file resides on a tftp server.

The `<src-full-path>` variable specifies the full directory path and name of the file that you want to copy to the USB drive.

The `<tftp-server-ip-addr>` parameter is the IP address of the tftp server. Note that this can be an IPV6 address (X:X::X:X).

The `<dest-file>` parameter specifies the destination file name.

Deleting files on USB drives and code flash

Use the **delete** or **del** commands to delete specified files from either the internal USB drive (**usb0**) or a USB drive attached to the external USB port (**usb1**).

Following are examples of deleting files on a USB drive and in code flash:

- Use the following command to delete the file named "asm12000.bin" from the root level of a USB drive attached to the USB (**usb1**) port of the ServerIron ADX.

```
ServerIronADX# delete usb1/asm12000.bin
```
- Use the following command to delete the file name "asm1000.bin" from the first-level directory "dir1" on the internal USB drive.

```
ServerIronADX# delete usb0/dir1/asm12000.bin
```
- Use the following command to delete a file named "primary" located in code flash.

```
ServerIronADX# delete primary
```
- Use the following command to delete the file named "filexxx.bin" from the second-level directory "dir2" on the external drive attached to the USB port (**usb1**) on the ServerIron ADX.

```
ServerIronADX# delete usb1/dir1/dir2/filexxx.bin
```

Syntax: **delete** `<file-full-path>`

or

Syntax: **del** `<file-full-path>`

The **delete** or **del** command deletes the specified file.

The `<file-full-path>` parameter specifies the full directory path and name of the file that you want to delete. The `<file-full-path>` must start with **usb0** or **usb1**, otherwise the file is assumed to be in code flash.

Renaming files on USB drives and code flash

Use the **rename** command to rename a specific file on either the internal USB drive (**usb0**), a USB drive attached to the external USB port (**usb1**), or in code flash.

Following are examples of renaming files on USB drives or in code flash:

- Use the following command to rename the file "filexxx.bin" in the first-level directory "dir1" of the internal USB drive (**usb0**) to the name "fileyyy.bin".

```
ServerIronADX# rename usb0/dir1/filexxx.bin fileyyy.bin
```

- Use the following command to rename the file "asm12000.bin" in the root level of a USB drive attached to the external USB port (**usb1**) of the ServerIron ADX to "asm12000b.bin".

```
ServerIronADX# rename usb1/asm12000.bin asm12000b.bin
```

- Use the following command to rename the file "file_abc.bin" in the second-level directory "dir2" of the USB drive attached to the external USB port (**usb1**) of the ServerIron ADX to the name "file_xyz.bin".

```
ServerIronADX# rename usb1/dir1/dir2/file_abc.bin file_xyz.bin
```

- Use the following command to rename the secondary image in code flash to sec2.

```
ServerIronADX# rename secondary sec2
```

Syntax: `rename <file-full-path> <new-filename>`

The *<file-full-path>* parameter specifies the full directory path and name of the file that you want to delete. The *<file-full-path>* must start with **usb0** or **usb1**, otherwise the file is assumed to be in code flash.

The *<new-filename>* parameter specifies new name for the file specified in the *<file-full-path>* parameter.

Formatting USB drives

You can format either the internal USB drive (**usb0**) or a USB drive attached to the external USB port (**usb1**) with the fat32 file system. The following example formats a USB drive attached to the USB port of the ServerIron ADX with the fat32 file system.

NOTE

Formatting a USB drive deletes all data on that drive. When you use this command, you will be prompted with this information before proceeding.

For **ADX 1U/4U** and **10U with single management module**, enter the following command.

```
ServerIronADX# usb format 1
```

Syntax: `usb format 0 | 1`

The **0** parameter directs the ServerIron ADX to format its internal USB drive.

The **1** parameter directs the ServerIron ADX to format an externally connected USB drive.

For a **10U with DUAL management module**, enter the following command.

```
ServerIronADX# usb format 0 both
```

Syntax: `usb format 0 <active-only | standby-only | both>`

The **0** parameter directs the ServerIron ADX to format its internal USB drive.

The *<active-only | standby-only | both>* parameter specifies whether the MP is in active mode, standby mode or both. The default value is "both".

NOTE

The help prompt (?) will display the *<active-only | standby-only | both>* parameter only when the redundant MPs and standby MPs exist in the device.

1 Clearing persistent information before an RMA

Testing USB Drives

You can test either the internal USB drive (**usb0**) or a USB drive attached to the external USB port (**usb1**). The following example tests a USB drive attached to the USB port of the ServerIron ADX.

```
ServerIronADX# usb test 1
```

Syntax: **usb test 0 | 1**

The **0** parameter directs the ServerIron ADX to test the internal USB drive.

The **1** parameter directs the ServerIron ADX to test an externally connected USB drive.

Clearing persistent information before an RMA

The following commands have been provided to allow you to delete persistent information from a ServerIron ADX before RMA.

NOTE

To delete all information on the internal USB drive, you can use the `usb format 0` command as described in ["Formatting USB drives"](#) on page 29.

Erasing flash data

You can delete all files in the code flash as shown in the following.

```
ServerIronADX# erase flash all-data
```

Syntax: **erase flash all-data** *<active-only | standby-only | both>*

The *<active-only | standby-only | both>* parameter specifies whether the MP is in active mode, standby mode or both. The default value is "both".

NOTE

The help prompt (?) will display the *<active-only | standby-only | both>* parameter only when the redundant MPs and standby MPs exist in the device.

Clearing crash dump Information

Management module crash dump files which are not visible to users can be deleted as shown in the following.

```
ServerIronADX# clear mp-dumps
```

Syntax: **clear mp-dumps**

Configuring SNTP

Simple Network Time Protocol (SNTP) ensures all devices have a synchronized time and date. If the ServerIronADX is configured to reference an authoritative SNTP server, the ServerIronADX automatically sets its system time counter according to the server (even after a system reset). Refer to RFC 1769 for more information. Refer to **show clock** to display the current settings.

Configuring an SNTP server location

You can define the SNTP server's location and specify an IP address or hostname. You can configure up to three SNTP servers by entering three separate **sntp server** commands.

To configure an SNTP server location, enter a command such as the following.

```
ServerIronADX(config)# sntp server 1.1.1.1
```

Syntax: **[no] sntp server** <ip-addr> | <hostname> [<version>]

The <version> parameter specifies the SNTP version the server is running and can be from 1 – 4. The default is 1.

Defining how often the clock references are validated

To define how often the clock references are validated between the devices, enter a command such as the following.

```
ServerIronADX(config)# sntp poll-interval 3
```

Syntax: **[no] sntp poll-interval** <value>

The default <value> is 1800 seconds.

Synchronizing the system clock

To manually synchronize the ServerIronADX's system clock with the time supplied by the SNTP server, enter the following command.

```
ServerIronADX# sntp sync
```

Syntax: **sntp sync**

Displaying SNTP information

To verify communications, enter the following command.

```
ServerIronADX# show sntp status
Clock is unsynchronized, stratum = 0, no reference clock
precision is 2**0
reference time is 0.0
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 0.00 msec, peer dispersion is 0.00 msec
sntp poll-interval is 3 secs

ServerIronADX#show sntp associations
  address          ref clock      st  when  poll  delay  disp
~1.1.1.1          0.0.0.0          16 51310    0    0.0    0.0
* synced, ~ configured
```

Syntax: **show sntp**

The following table describes the information displayed by the **show sntp status** command.

This field...	Indicates...
unsynchronized	System is not synchronized to an NTP peer.
synchronized	System is synchronized to an NTP peer.
stratum	NTP stratum level of this system
reference clock	IP Address of the peer (if any) to which the unit is synchronized
precision	Precision of this system's clock (in Hz)
reference time	Reference time stamp
clock offset	Offset of clock to synchronized peer
root delay	Total delay along the path to the root clock
root dispersion	Dispersion of the root path
peer dispersion	Dispersion of the synchronized peer

To display information about SNTP associations, enter the following command.

```
ServerIronADX# show sntp associations
  address      ref clock      st  when  poll  delay  disp
~207.95.6.102  0.0.0.0          16  202    4    0.0    5.45
~207.95.6.101  0.0.0.0          16  202    0    0.0    0.0
* synced, ~ configured
```

Syntax: show sntp associations

The following table describes the information displayed by the **show sntp associations** command.

This field...	Displays...
(leading character)	One or both of the following: * Synchronized to this peer ~ Peer is statically configured
address	IP address of the peer
ref clock	IP address of the peer's reference clock
st	NTP stratum level of the peer
when	Amount of time since the last NTP packet was received from the peer
poll	Poll interval in seconds
delay	Round trip delay in milliseconds
disp	Dispersion in seconds

Configuring DNS

This section contains information on configuring DNS.

Defining a domain name

You can define a domain name for a range of addresses on the ServerIron ADX. This will eliminate the need for a user to type in the domain name. It will automatically be appended to the hostname.

To define a domain name, enter a command such as the following.

```
ServerIronADX(config)# ip dns domain-name brocade.com
```

Syntax: [no] ip dns domain-name <name>

Defining DNS servers

You can define up to four DNS servers for each DNS entry. The first entry serves as the primary default address (207.95.6.199). If a query to the primary address fails to be resolved after three attempts, the next gateway address will be queried for three times as well. This process will continue for each defined gateway address until a query is resolved. The order in which the default gateway addresses are polled is tied to the order in which they are entered when initially defined as shown in the example.

To define DNS servers, enter a command such as the following.

```
ServerIronADX(config)#ip dns server-address 207.95.6.199 205.96.7.1 208.95.7.25
```

Syntax: [no] ip dns server-address <ip-addr>

Configuring DNS Resolver

The Domain Name Server (DNS) Resolver feature allows you to use just a host name rather than a fully-qualified domain name when you use Telnet, ping, and trace-route commands.

To configure the feature, you specify the domain name, then specify the IP addresses of up to four DNS servers that have authority for the domain.

For example, if you define the domain “example.com” on a Brocade device, you can initiate a ping to a host on that domain by specifying only the host name in the command. You do not need to specify the host’s entire domain name.

As an example, here are two CLI commands.

```
ServerIronADX# ping nyc01  
ServerIronADX# ping nyc01.example.com
```

The first command uses only the host name. The second command uses the fully-qualified domain name for the host.

Configuring SNMP

This section contains information on configuring SNMP.

SNMP support

Simple Network Management Protocol (SNMP) version 1 and SNMPv2c are enabled by default and cannot be disabled. For SNMPv3 you need to enable the device to process SNMPv3 packets.

Traps

To display a subset of the supported traps, use the **show snmp server** command. Some of the traps cannot be disabled. Consult the related MIB for more information.

Partial trap list:

- **SNMP Authentication** – Indicates a failed attempt to access the device through SNMP using an invalid SNMP community string.
- **Power Supply** – Indicates a power supply failure.
- **Fan** – Indicates a fan failure.
- **Cold Start** – Indicates a restart from a powered down state.
- **Link Up** – Indicates that a port link has come up.
- **Link Down** – Indicates that a port link has gone down.
- **Bridge New Root** – Indicates a spanning-tree change.
- **Bridge Topology Change** – Indicates a spanning-tree change.
- **Lock Address Violation** – Indicates that a locked port received a packet from a MAC address that is not allowed access to that port.
- **Maximum Session** – Indicates that the maximum number of sessions has been reached. A session is either a send or receive link between the ServerIron ADX and a real server. Two sessions make a two-way connection between the ServerIron ADX and a server.
- **TCP SYN Limit** – Indicates that the maximum TCP SYN rate has been reached on a real server.
- **Real Server Max Connection** – Indicates that a real server has reached the maximum number of connections the ServerIron ADX is configured to allow on that server. A connection represents both the receive and send sessions.
- **Real Server Up** – Indicates that a real server has come up.
- **Real Server Down** – Indicates that a real server has gone down.
- **Real Server Port Up** – Indicates that a port on a real server has come up.
- **Real Server Port Down** – Indicates that a port on a real server has gone down.
- **Cache Server Up** – Indicates that a cache server has come up.
- **Cache Server Down** – Indicates that a cache server has gone down.
- **Cache Server Port Up** – Indicates that a TCP port on a cache server has come up.
- **Cache Server Port Down** – Indicates that a TCP port on a cache server has gone down.
- **Switch Standby** – Indicates that an SLB switch fail-over has occurred, and the active switch is down.
- **Switch Active** – Indicates that the standby switch is active.

All traps are enabled by default.

NOTE

You can disable SNMP access to the device if needed.

NOTE

IronView Network Manager (INM) supports SNMP V1/V2c/V3 on UNIX and Windows. Refer to the IronView Network Manager User guide.

The following enterprise trap generated by a ServerIron ADX has been enhanced to display the port name and the port number in the trap message. Previously, this message displayed port number only:

- snTrapLockedAddressViolation2(32)

This trap is generated when the number of source MAC addresses received from a port is greater than the maximum number of MAC addresses configured for that port. It displays the following trap message.

```
Locked address violation at <port-name> <port-num>, address <mac>
```

In addition, the following standard traps now display the port name and port number in the trap message when generated by the ServerIron ADX. Previously, these messages displayed port number only:

- linkDown(2)

This trap is generated when a port state changes to DOWN. It displays the following trap message.

```
Interface <port-name> <port-num>, state down
```

- linkUp(3)

This trap is generated when a port state changes to UP. It displays the following trap message.

```
Interface <port-name> <port-num>, state up
```

NOTE

The trap receiver you are using determines whether or not port name and port number is displayed. If you are using IronView as the trap receiver, port name and port number are displayed. Contact Brocade for more information.

Using the MIB table

The Real Server Port Statistics MIB table (snL4RealServerPortStatisticTable) has been updated to include information for remote servers. Previously, objects in this table displayed information only for real servers. The OID for snL4RealServerPortStatistic table is 1.3.6.1.4.1.1991.1.1.4.24.1.

The following object is added to the snL4RealServerPortCfgTable (Real Server Port Configuration Table).

Name, OID, and Syntax	Access	Description
snL4RealServerPortCfgMaxConnections fdry.1.1.4.20.1.1.7 Syntax: Integer	Read-write	Defines the maximum number of connections allowed per port. Enter a value up to 1000000.

Restricting SNMP management access

You can restrict SNMP management access to the ServerIron ADX to the host whose IP address you specify. No other device except the one with the specified IP address can access the Brocade device through IronView or any other SNMP application.

If you want to restrict access from Telnet or the Web, use one or two of the following commands:

- **telnet client** – restricts Telnet access.
- **web client** – restricts Web access.

If you want to restrict all management access, you can use the commands above and the **snmp-client** command or you can use the **all-client** command.

To restrict SNMP access (which includes IronView) to the Brocade device to the host with IP address 209.157.22.26, enter the following command.

```
ServerIronADX(config)# snmp-client 209.157.22.26
```

Syntax: [no] snmp-client <ip-addr>

You can use the command up to ten times for up to ten IP addresses.

Assigning an SNMP community string

You can assign an SNMP community string for the system. It will register to the configuration file, a user-specified network community string and an access type of either:

- read-only (public)
- read-write (private)

The default read-only community string is "public". There is no default read-write community string.

In ServerIron, the global command **snmp-server** is enabled by default and this command must not be removed if the dynamic weighted predictor is configured. If this command is removed, then the ServerIron will stop listening on the UDP port 161 and drop SNMP responses from the real servers that are used for this predictor.

To assign an SNMP community string, enter a command such as the following.

```
ServerIronADX(config)# snmp-server community planet1 ro
```

Syntax: [no] snmp-server community <string> ro | rw

The <string> parameter can be up to 32 alphanumeric characters for the community string.

Designating a contact

You can designate a contact name for the ServerIron ADX and save it in the configuration file for later reference. You can later access contact information using the **show snmp server** command.

To identify a system contact, enter a command such as the following.

```
ServerIronADX(config)# snmp-server contact Noi Lampa
```

Syntax: [no] snmp-server contact <text>

The <text> parameter can be up to 32 alphanumeric characters for the system contact text string.

Enabling or disabling traps

By default, all of the following SNMP traps are enabled and will be generated by default for a system: authentication key, cold-start, link-up, link-down, new-root, topology-change, power-supply-failure and locked-address-violation.

You can use the **snmp-server enable traps <name>** command to enable other trap types, such as I4-port-down and I4-port-up.

When the command is preceded with the word **no**, the command is used to stop certain traps from being generated by a system. To disable a fan failure trap or power supply trap, use one of the following values: ps1 | ps2 | ps3 | ps4 | fan1 | fan2 | fan3 | fan4.

To stop reporting incidences of links that are down, enter the following command.

```
ServerIronADX(config)# no snmp-server enable traps link-down
```

Syntax: [no] snmp-server enable traps <name>

Allowing SNMP access only to clients in a VLAN

You can allow SNMP access only to clients in a specific VLAN.

The following example configures the device to allow SNMP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

```
ServerIronADX(config)# snmp-server enable vlan 40
```

Syntax: [no] snmp-server enable vlan <vlan-id>

Enabling or disabling a station as an SNMP trap receiver

You can assign or remove a station as SNMP trap receiver. To assign the trap receiver, use the command: **snmp-server host**. To later remove the trap receiver feature, enter **no snmp-server host**.

To disable a station as a SNMP trap receiver, enter a command such as the following.

```
ServerIronADX(config)# no snmp-server host 192.22.3.33 public
```

Syntax: [no] snmp-server host <ip-addr-of-trap-receiver-station> <community-string>

Identifying a system location

You can identify a system location for the ServerIron ADX. This information is saved in the configuration file for later reference. You can later access system location information using the **show snmp server** command.

To identify a system location, enter a command such as the following.

```
ServerIronADX(config)# snmp-server location pulchritude_lane
```

Syntax: [no] snmp-server location <text>

Disabling password checking

You can disable password checking for SNMP set requests. If a third-party SNMP management application does not add a password to the password field when it sends SNMP set requests to a Brocade device, by default the Brocade device rejects the request.

To disable password checking, enter the following command.

```
ServerIronADX(config)# no snmp-server pw-check
```

Syntax: [no] snmp-server pw-check

Specifying the source for all SNMP traps

You can specify a port or virtual interface whose lowest-numbered IP address the Brocade device must use as the source for all SNMP traps sent by the device. To do so, enter a command such as the following.

```
ServerIronADX(config)# snmp-server trap-source ethernet 4
```

Syntax: [no] snmp-server trap-source ethernet <portnum> | ve <num>

The **ethernet** <portnum> parameter specifies a physical port on the device. Alternatively, you can specify a virtual interface using the **ve** <num> parameter, where <num> is the number of a virtual interface configured on the device. It uses the lowest-numbered address on the interface.

NOTE

The command **snmp-server trap-source** will change the source IP address used for syslog messages to a syslog server.

Configuring an SNMP view

You can use an SNMP view as an argument with other commands.

SNMP views are named groups of MIB objects that can be associated with user accounts to allow limited access for viewing and modification of SNMP statistics and system configuration. SNMP views can also be used with other commands that take SNMP views as an argument. SNMP views reference MIB objects using object names, numbers, wildcards, or a combination of the three. The numbers represent the hierarchical location of the object in the MIB tree. You can reference individual objects in the MIB tree or a subset of objects from the MIB tree.

To configure an SNMP view, enter commands such as the following.

```
ServerIronADX(config)# snmp-server view Maynes system included
ServerIronADX(config)# snmp-server view Maynes system.2 excluded
ServerIronADX(config)# snmp-server view Maynes 2.3.*.6
ServerIronADX(config)# write mem
```

Syntax: [no] snmp-server view <name> <mib_tree> included | excluded

The <name> parameter can be any alphanumeric name you choose to identify the view. The names cannot contain spaces.

The <mib_tree> parameter is the name of the MIB object or family. MIB objects and MIB sub-trees can be identified by name or by the numbers representing the position of the object or sub-tree in the MIB hierarchy. You can use a wildcard (*) in the numbers to specify a sub-tree family.

The **included | excluded** parameter specifies whether the MIB objects identified by the <mib_family> parameter are included in the view or excluded from the view.

NOTE

The **snmp-server view** command supports the MIB objects as defined in RFC 1445.

Clearing all statistics for SNMP server traffic

To clear all statistics for SNMP server traffic, enter the following command.

```
ServerIronADX# clear snmp-server traffic
```

Syntax: clear snmp-server traffic

Configuring access control

This section contains information on configuring Access Control.

Enabling configuration of RADIUS

To enable users of IronView or other SNMP management applications to configure RADIUS authentication parameters on the ServerIron ADX, enter the following command.

```
ServerIronADX(config)# enable snmp config-radius
```

Syntax: [no] enable snmp config-radius

Enabling configuration of TACACS or TACACS+

To enable users of IronView or other SNMP management applications to configure TACACS or TACACS+ authentication parameters on the ServerIron ADX.

```
ServerIronADX(config)#enable snmp config-tacacs
```

Syntax: [no] enable snmp config-tacacs

Restricting management access to the ServerIronADX

You can restrict management access to the ServerIronADX. No other host except the one with the IP address you specify can access the ServerIronADX through Telnet (CLI), the Web (Web Management Interface), or SNMP (IronView). Replace *<ip-addr>* with a valid IP address. You can enter one IP address with the command, but you can issue the command up to ten times for up to ten IP addresses.

If you want to restrict access for some of the management platforms but not all of them, use one or two of the following commands:

- **snmp-client** – restricts IronView access and all other SNMP access.
- **telnet client** – restricts Telnet access.
- **web client** – restricts web access.

To restrict management access to the ServerIronADX, enter the following command.

```
ServerIronADX(config)# all-client 209.157.22.26
```

Syntax: [no] all-client *<ip-addr>*

Determining the access points where the password can be defined

To define the access points from which the system password can be defined, enter a command such as the following.

```
ServerIronADX(config)# password-change cli
```

Syntax: [no] password-change [any | cli | console-cli | telnet-cli]

The **any** option would allow the password to be modified from a serial port, telnet session, or through IronView.

Enhancing access privileges

You can augment the default access privileges for an access level. When you configure a user account, you can give the account one of three privilege levels: full access, port-configuration access, and read-only access. Each privilege level provides access to specific areas of the CLI by default:

- Full access provides access to all commands and displays.
- Port-configuration access gives access to:
The User EXEC and Privileged EXEC levels, and the port-specific parts of the CONFIG level
All interface configuration levels
- Read-only access gives access to:
The User EXEC and Privileged EXEC levels

To enhance the port-configuration privilege level so users also can enter ip commands at the global CONFIG level (useful for adding IP addresses for multinetting), enter a command such as the following.

```
ServerIronADX(config)#privilege configure level 4 ip
```

This command specifies that the enhanced access is for a command at the global CONFIG level of the CLI. The level 4 parameter indicates that the enhanced access is for privilege level 4 (port-configuration). All users with port-configuration privileges will have the enhanced access. The ip parameter indicates that the enhanced access is for the IP commands. Users who log in with valid port-configuration level user names and passwords can enter commands that begin with "ip" at the global CONFIG level.

Syntax: **[no] privilege** *<cli-level>* **level** *<privilege-level>* *<command-string>*

The *<cli-level>* parameter specifies the CLI level and can be one of the following values:

exec – EXEC level; for example, ServerIronADX> or ServerIronADX#

configure – CONFIG level; for example, ServerIronADX(config)#

interface – interface level; for example, ServerIronADX(config-if-6)#

port-vlan – port-based VLAN level; for example, ServerIronADX(config-vlan)#

protocol-vlan – protocol-based VLAN level; for example, ServerIronADX(config-vlan)#

The *<privilege-level>* parameter indicates the privilege level you are augmenting.

The **level** parameter specifies the privilege-level. You can specify one of the following:

- 0 – Full access (super-user)
- 4 – Port-configuration access
- 5 – Read-only access

The *<command-string>* parameter specifies the command you are allowing users with the specified privilege level to enter. To display a list of the commands at a CLI level, enter "?" at that level's command prompt and press Return.

TACACS and TACACS+

You can secure CLI access to the switch or router by configuring the device to consult a Terminal Access Controller Access Control System (TACACS) or TACACS+ server to authenticate user names and passwords.

NOTE

TACACS or TACACS+ authentication is not supported for Web management or IronView access.

Setting TACACS or TACACS+ parameters

To identify a TACACS or TACACS+ server and set other TACACS or TACACS+ parameters for authenticating access to the ServerIronADX, enter a command such as the following.

```
ServerIronADX(config)# tacacs-server host 209.157.22.99
```

Syntax: [no] tacacs-server host <ip-addr> | <server-name> [auth-port <number>]

Syntax: [no] tacacs-server [key <key-string>] [timeout <number>] [retransmit <number>] [dead-time <number>]

The only required parameter is the IP address or host name of the server. To specify the server's host name instead of its IP address, you must first identify a DNS server using the **ip dns server-address <ip-addr>** command at the global CONFIG level.

The **auth-port** parameter specifies the UDP port number of the authentication port on the server. The default port number is 49.

The **key** parameter specifies the value that the Brocade device sends to the server when trying to authenticate user access. The TACACS or TACACS+ server uses the key to determine whether the Brocade device has authority to request authentication from the server. The key can be from 1 – 16 characters in length.

The **timeout** parameter specifies how many seconds the Brocade device waits for a response from the TACACS or TACACS+ server before either retrying the authentication request or determining that the TACACS or TACACS+ server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

The **retransmit** parameter specifies how many times the Brocade device will re-send an authentication request when the TACACS or TACACS+ server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.

When the software allows multiple authentication servers, the **dead-time** parameter specifies how long the Brocade device waits for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 – 5 seconds. The default is 3.

Enabling command authorization and accounting at the console

To configure the device to perform command authorization and command accounting for commands entered at the console, enter the following command.

```
ServerIronADX(config)# enable aaa console
```

Syntax: [no] enable aaa console

ATTENTION

If you have previously configured the device to perform command authorization using a RADIUS server, entering **enable aaa console** may prevent the execution of any subsequent commands entered on the console.

This happens because RADIUS command authorization requires a list of allowable commands from the RADIUS server. This list is obtained during RADIUS authentication. For console sessions, RADIUS authentication is performed only if you have configured Enable authentication and specified RADIUS as the authentication method (for example, with **aaa authentication enable default radius**). If RADIUS authentication is never performed, the list of allowable commands is never obtained from the RADIUS server. Consequently, there would be no allowable commands on the console.

Displaying information about TACACS+ and RADIUS servers

To display information about all TACACS+ and RADIUS servers identified on the device, enter the following command.

```
ServerIronADX# show aaa
Tacacs+ key: brocade
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                  opens=6 closes=3 timeouts=3 errors=0
                  packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server: 207.95.6.90 Auth Port=1645 Acct Port=1646:
                  opens=2 closes=1 timeouts=1 errors=0
                  packets in=1 packets out=4
no connection
```

Syntax: show aaa

RADIUS security

You can further secure CLI access to the switch or router by configuring the device to consult a RADIUS server to authenticate user names and passwords. You can configure the device to authenticate Telnet logins and Enable access on a separate basis.

NOTE

RADIUS authentication is not supported for Web management or IronView access.

Setting RADIUS server parameters

You can identify a RADIUS server and sets other RADIUS parameters, by entering a command such as the following.

```
ServerIronADX(config)# radius-server host 209.157.22.99
```

Syntax: [no] radius-server host <ip-addr> | <server-name> [auth-port <number>] [acct-port <number>]

Syntax: [no] radius-server [key <key-string>] [timeout <number>] [retransmit <number>] [dead-time <number>]

The <ip-addr> | <server-name> parameter specifies either an IP address or an ASCII text string.

The optional <auth-port> parameter specifies Authentication port number. The default is 1645.

The optional <acct-port> parameter specifies the accounting port number. The default is 1646.

The <key-string> parameter specifies the encryption key. Valid key string length is from 1 – 16.

The **timeout** <number> parameter specifies how many seconds to wait before declaring a RADIUS server timeout for the authentication request. The default timeout is 3 seconds. The range of possible timeout values is from 1 – 15.

The **retransmit** <number> parameter specifies the maximum number of retransmission attempts. When an authentication request timeout, the Brocade software will retransmit the request up to the maximum number of retransmissions configured. The default retransmit value is 3 seconds. The possible retransmit value is from 1 – 5.

When the software allows multiple authentication servers, the **dead-time** parameter specifies how long the Brocade device wait for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 – 5 seconds. The default is 3.

Password recovery

By default, the CLI does not require passwords. However, if someone has configured a password for the ServerIron ADX but the password has been lost, you can regain super-user access to the ServerIron ADX using the following procedure.

NOTE

Recovery from a lost password requires direct access to the serial port and a system reset.

Follow the steps listed below to recover from a lost password.

1. Start a CLI session over the serial interface to the ServerIron ADX.
2. Reboot the ServerIron ADX.
3. While the system is booting, before the initial system prompt appears, enter b to enter the boot monitor mode.
4. Enter **no password** at the prompt. (You cannot abbreviate this command.)
5. Enter **boot system flash primary** at the prompt. This command causes the device to bypass the system password check.
6. After the console prompt reappears, assign a new password.

Displaying information about the security feature

To display which security features are enabled on the system, enter the following command.

```
ServerIronADX1/1#show feature
17 switching          : OFF
sFlow                 : OFF
NAT                   : ON
TCS/FW               : OFF
ACL                   : OFF
inbound ACL          : OFF
GSLB controller      : ON
SYN proxy             : ON
SYN defence           : OFF
SLB only              : OFF
```

Syntax: show feature

Power budgeting on the ServerIron ADX

The following power budgeted it available on the ServerIron ADX models as shown in the following:

- **ServerIron ADX 1000** – A maximum of 2 power supplies are available. Each power supply is rated at 504 W. A single power supply will meet the demands of a fully operating ServerIron ADX 1000. The ServerIron ADX 1000 is not subject to power budgeting.
- **ServerIron ADX 4000** – A maximum of 2 power supplies are available. At least one power supply must be connected and operating in a fully-loaded ServerIron ADX 4000 chassis.
- **ServerIron ADX 8000** – A maximum of 4 power supplies are available. At least two power supplies must be connected and operating in a fully-loaded ServerIron ADX 8000 chassis.

[Table 1](#) describes the amount of power in Watts required to bring-up the components of a ServerIron ADX 4000 or ServerIron ADX 8000 system.

TABLE 1 ServerIron ADX start-up power requirements

ServerIron ADX system component	Power requirements forStart-up
Interface Module	74 W
Switch Fabric Module	69 W
Application Switch Module	330 W
Management Module	140 W
Fans at 100% RPM	84 W

Operation of power budgeting

With ServerIron ADX 4000 and ServerIron ADX 4000 and ServerIron ADX 8000, the system follows a procedure for powering-up various components. If the power demands of a component exceed the power budget of the ServerIron ADX system, the component will not be brought up and a message will be sent to the SYSLOG. The sequence for power-up is described in the following steps.

1. The power required to operate the Management module (or modules if a standby Management module is installed) and the fans (at 100% RPM) is deducted from the available power budget.

NOTE

The power budget is calculated based on the number of power supplies that are operating in the system and the power consumption of the system component is calculated using the values described in [Table 1](#).

2. The Interface Modules are powered-up starting with Slot 1 and continuing sequentially through all installed Interface modules. If the power budget is exceeded during this procedure a SYSLOG error message is generated.
3. The Switch Fabric Modules are powered-up in sequence starting with SF1. If the power budget is exceeded during this procedure a SYSLOG error message is generated.
4. The Application Switch Modules (ASM) are powered-up starting with Slot 1 and continuing sequentially through all installed ASM modules. If the power budget is exceeded during this procedure a SYSLOG error message is generated.

Configuring the cooling system

The ServerIron ADX switch has automatic fan speed control. The fans operate at the following speeds:

- **low** – 50% of the maximum RPM
- **med** – 75% of the maximum RPM
- **med_hi** – 90% of the maximum RPM
- **hi** – 100% of the maximum RPM

If any module exceeds a temperature threshold the fan speed is bumped-up to the next level. If all temperatures monitored in the chassis drop below a threshold, the fan speed is bumped down to the previous level.

[Table 2](#) provides the default low and high temperature thresholds for each module and the associated fan speed.

TABLE 2 Default low and high temperature thresholds for modules and fan speeds

Fan Speed	Low Temperature Threshold	High Temperature Threshold
Active Management module		
High	77° C	85° C
Medium-high	67° C	80° C
Medium	0° C	70° C
Low	-1°	60° C
Standby Management moduls		
High	77° C	95° C
Medium-high	67° C	80° C
Medium	0° C	70° C
Low	-1°	60° C

1 Configuring the cooling system

TABLE 2 Default low and high temperature thresholds for modules and fan speeds (Continued)

Fan Speed	Low Temperature Threshold	High Temperature Threshold
Interface modules		
High	57° C	75° C
Medium-high	47° C	60° C
Medium	0° C	50° C
Low	-1°	37° C
Switch fabric module		
High	57° C	75° C
Medium-high	47° C	60° C
Medium	0° C	50° C
Low	-1°	37° C
ASM module		
High	70° C	95° C
Medium-high	62° C	80° C
Medium	0° C	70° C
Low	-1°	60° C
ServerIron ADX 1000		
High	77° C	95° C
Medium-high	69° C	80° C
Medium	57° C	70° C
Low	-1°	60° C

To view the current temperatures of devices in the ServerIron ADX switch, refer to [“Displaying chassis information”](#) on page 56.

Setting a fan speed manually

You can manually set the speed for any or all fans in a ServerIron ADX switch using the following command.

```
ServerIronADX# fan-speed 0 hi
```

Syntax: [no] fan-speed <fan-number> lo | med | med-hi | hi

The <fan-number> variable specifies which fan you want to set the speed for. For the ServerIron ADX 8000 this value can be: 1 - 6. For the ServerIron ADX 4000 this value can be: 1 - 3. For the ServerIron ADX 1000 this value can be: 1 - 2. Selecting 0 sets the speed for all fans in the chassis.

The lo | med | med-hi | hi parameters set the fan speed as described in [“Configuring the cooling system”](#) on page 45.

Configuring a redundant management module

In a ServerIron ADX chassis that contains a redundant Management module,

You can install a redundant management module in a ServerIron ADX chassis. (By default, the system considers the module in the lower slot number to be the active management module and the other module to be the redundant, or standby module. If the active module becomes unavailable, the standby module automatically takes over management of the system. You can however override the default and make the redundant Management module in the higher slot number the default active module.

NOTE

This feature only applies for chassis-based ServerIron ADX switches that are equipped with redundant Management module.

To change the active management module in a ServerIron ADX chassis to the module in the higher slot number, use the following command.

```
ServerIronADX(config)# redundancy
ServerIronADX(config-redundancy)# active-management 2
ServerIronADX(config-redundancy)# exit
ServerIronADX(config)# exit
ServerIronADX# write-memory
ServerIronADX# reload
```

Syntax: `[no] active-management <module-number>`

The `<module-number>` variable specifies the management module that will assume Active management upon reboot of the ServerIron ADX switch. You can specify 1 or 2. If you specify 1, the default Management module will be active. If you specify 2, the redundant Management module in the higher slot number will be the Active module after reboot.

NOTE

The change in active Management module does not take effect until you reload the switch. If you save the change to the active module's system-config file before reloading, the change persists across system reloads. Otherwise, the change affects only the next system reload.

Synchronizing the active and standby modules

You can immediately synchronize software between the active and standby management modules. When you synchronize software, the active module copies the software you specify to the standby module, replacing the software on the standby module.

To immediately synchronize the configuration data (in flash memory or flash module) on the standby module with the configuration data on the active module, enter the following command.

```
ServerIronADX# sync-standby config
```

To immediately synchronize the flash image code (system software) on the standby module with the code image on the active module (in flash memory or flash module), enter the following command.

```
ServerIronADX# sync-standby code
```

To immediately synchronize the SSL keys and certificates on the standby module with the SSL keys and certificates on the active module, enter the following command.

```
ServerIronADX# sync-standby ssl-key-cert
```

Syntax: `sync-standby { code | config | ssl-key-cert }`

NOTE

The **sync-standby boot** command was deprecated in release 12.2.0 and is no longer available.

NOTE

The **sync-standby** command applies only to a ServerIron ADX with redundant management modules. BP boot and flash code must be synchronized manually. We recommend re-downloading over TFTP to simultaneously update the BP boot and flash images on both modules. To download software to both modules, use the BP boot and flash upgrade instructions in the release notes.

High availability configurations

This section provides detailed information for creating high-availability ServerIron ADX configurations.

Synchronizing the configurations

You can synchronize the configurations of ServerIron ADXs in a network by changing to the **configure sync-terminal** level. Commands entered at this level on one ServerIron ADX are duplicated on other ServerIron ADXs in the network where the following features are configured:

- Layer 4 Server Load Balancing features, for example, SLB, Symmetric SLB, hot standby redundancy
- Layer 7 Switching features such as URL switching, cookie switching, HTTP header hashing, and SSL session ID switching
- Health checks

NOTE

The “config-sync” feature is not supported on a ServerIron ADX when used in multiple context scenarios.

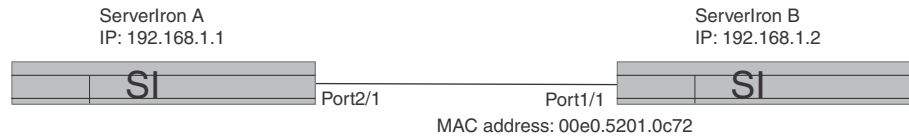
For example, in [Figure 3](#), the commands you enter on ServerIron ADX A while that device is at the **configure sync-terminal** level, are duplicated on ServerIron ADX B.

FIGURE 3 ServerIron ADXs with connection to each other



In Figure 3, the commands you enter on ServerIron ADX A while that device is at the **configure sync-terminal** level, are duplicated on ServerIron ADX B.

FIGURE 4 ServerIron ADXs with connection to each other



Preparing for synchronization

To be able to send commands to other ServerIron ADXs, do the following before entering the **configure sync-terminal** level:

- Make sure the physical ports used to connect the ServerIron ADXs are enabled.
- Make sure that you have enabled the **config-sync mac** command on the source ServerIron ADX (the ServerIron ADX where the commands will be entered). Do not enter the **config-sync mac** command on the destination ServerIron ADXs (the ServerIron ADXs where the configuration will be duplicated).

NOTE

If you configure **config-sync mac** on both ServerIron ADXs, this feature will not work.

The **config-sync** command indicates on which port on the source ServerIron ADX the commands will be sent. It also indicates the destination MAC address or VLAN ID on the destination ServerIron ADX. For example, if you are configuring hot standby on ServerIron ADX A and those commands will be duplicated on ServerIron ADX B, begin the synchronization by entering commands such as the following on ServerIron ADX A.

```
ServerIronADXA# configure terminal
ServerIronADXA(config)# config-sync eth 2/1 mac 00e0.5201.0c72
ServerIronADXA(config)# write mem
ServerIronADXA(config)# exit
ServerIronADXA#
```

Syntax: **config-sync ethernet** <port-number> **mac** <mac-address> [**vlan** <vlan-ID>]

The **ethernet** <port-number> parameter indicates which port on the source ServerIron ADX will be used to send the commands.

The **mac** <mac-address> parameter indicates the destination port of the commands on the destination ServerIron ADX.

The **vlan** <vlan-ID> parameter indicates the destination VLAN of the commands on the destination ServerIron ADX.

Initiating and ending the synchronization

Once you have indicated the port on which the commands will be sent and the destination MAC address or VLAN ID, you can begin to synchronize the configuration by entering commands such as the following on the source ServerIron ADX.

```
ServerIronADX# configure terminal
ServerIronADX(config)# config-sync eth 2 mac 00e0.5201.0c72
ServerIronADX(config)# write mem
ServerIronADX(config)# exit
ServerIronADX# configure sync-terminal
ServerIronADX(config-sync)# server virtual v1 10.10.1.1
ServerIronADX(config-sync-vs-v1)# port http
ServerIronADX(config-sync-vs-v1)# exit
ServerIronADX(config-sync)# write mem
ServerIronADX(config-sync)# exit
```

Syntax: configure sync-terminal

This command allows you to enter the **configure sync-terminal** level. Once at that level, you can configure the features in the following categories:

- Layer 4 Server Load Balancing features, for example, SLB, Symmetric SLB, hot standby redundancy
- Layer 7 Switching features such as URL switching, cookie switching, HTTP header hashing, and SSL session ID switching
- Health checks

Enter a "?" at the CLI command line to display the list of commands allowed for synchronization.

Commands entered on ServerIron ADX A while at the **configure sync-terminal** level are duplicated on ServerIron ADX B. The commands continue to be duplicated on ServerIron ADX B until you exit out of the **configure sync-terminal** level.

If you enter an invalid command while you are at the **configure sync-terminal** level or if the command you entered cannot be accepted by the destination ServerIron ADX, a message appears.

Example

```
Peer message: real server 1 not found
```

Also, some commands, such as **no server real**, may take a few seconds to process before the CLI is available for the next command.

To see if the configuration entered at the **configure sync-terminal** level was successfully duplicated on the destination ServerIron ADX, enter the **show run** command on the destination ServerIron ADX. The duplicated configuration should be displayed.

Creating config-sync peers

To be able to send commands to other ServerIron ADXs, do the following before entering the **configure sync-terminal** level:

- Ensure the physical ports used to connect the ServerIron ADXs are enabled.
- Enter the **config-sync sender** command on the source ServerIron ADX (the ServerIron ADX where the commands will be entered), and enter the **config-sync receiver** command on the destination ServerIron ADX (the ServerIron ADX where the commands will be received).

The **config-sync sender** command indicates the port on the source ServerIron ADX on which the commands will be sent. The command also indicates the destination MAC address or VLAN ID on the destination ServerIron ADX.

The **config-sync receiver** command enables the destination ServerIron ADX to receive configuration commands from the source ServerIron ADX. You can configure this command to allow the destination ServerIron ADX to receive configuration commands only on a specified port, MAC address, or VLAN ID. For added security, Brocade recommends that you establish a dedicated link between the source and destination ServerIron ADXs, in addition to specifying a source port for receiving configuration commands.

For example, if you are setting up a hot-standby configuration with the commands on ServerIron ADX A to be duplicated on ServerIron ADX B, begin the synchronization by entering commands such as the following on ServerIron ADX A.

```
ServerIronADXA# configure terminal
ServerIronADXA(config)# config-sync sender e 2/1 mac 00e0.5201.0c72
ServerIronADXA(config)# write mem
ServerIronADXA(config)# exit
```

Syntax: **config-sync sender ethernet** *<port-number>* **mac** *<mac-address>* [**vlan** *<vlan-id>*]

The **ethernet** *<port-number>* parameter indicates which Ethernet port on the source ServerIron ADX will be used to send the commands.

The **mac** *<mac-address>* parameter indicates the destination port for the commands on the destination ServerIron ADX.

The **vlan** *<vlan-id>* parameter indicates the destination VLAN for the commands on the destination ServerIron ADX.

Next, enter commands such as the following on ServerIron ADX B. The commands in this example allow the ServerIron ADX to receive configuration commands only from Ethernet port 1/1 with VLAN ID 5.

```
ServerIronADXB# configure terminal
ServerIronADXB(config)# config-sync receiver ethernet 1/1 vlan-id 5
ServerIronADXB(config)# write mem
ServerIronADXB(config)# exit
```

Syntax: **config-sync receiver ethernet** *<port-number>* | **any** **vlan** *<vlan-ID>* [**mac** *<mac-address>*]

The **ethernet** *<port-number>* | **any** parameter indicates the port from which the ServerIron ADX can receive configuration commands. Specify **any** to allow configuration commands to be received from any port.

The **vlan** *<vlan-id>* parameter indicates the VLAN from which the ServerIron ADX can receive configuration commands. If no VLANs are configured, enter the default VLAN ID.

The optional **mac** *<mac-address>* parameter indicates the source MAC address from which the ServerIron ADX can receive configuration commands. If you specify this parameter, enter the same MAC address you entered in the **config-sync sender** command on the source ServerIron ADX.

Initiating the synchronization

Once you have indicated the port on which the commands will be sent and the destination MAC address or VLAN ID, you can begin to synchronize the configuration by entering commands such as the following on the source ServerIron ADX.

```
ServerIronADXA# configure terminal
ServerIronADXA(config)# config-sync eth 2/1 mac 00e0.5201.0c72
ServerIronADXA(config)# write mem
ServerIronADXA(config)# exit
```

1 High availability configurations

```
ServerIronADXA# configure sync-terminal
ServerIronADXA(config-sync)# server virtual v1 10.10.1.1
ServerIronADXA(config-sync-vs-v1)# port http
ServerIronADXA(config-sync-vs-v1)# exit
ServerIronADXA(config-sync)# write mem
ServerIronADXA(config-sync)# exit
```

Syntax: configure sync-terminal

Once you enter the **configure sync-terminal** level, commands entered on ServerIron ADX A are duplicated on ServerIron ADX B. The commands continue to be duplicated on ServerIron ADX B until you **exit** out of the **configure sync-terminal** level.

Enter a “?” at the CLI command line to display the list of commands allowed for synchronization. This list can vary from device to device and from release to release.

```
ServerIronADX(config)# config-sync sender e 2/21 mac 000c.db2b.ad34
ServerIronADX(config)# exit
ServerIronADX# config sync-terminal
ServerIronADX(sync-config)# ?
  agent-health-report-interval
  csw-policy                    content switching policy name
  csw-rule                     content switching rule
  end                          End Configuration level and go to Privileged Level
  exit                         Exit current level
  extern-config-file           extern configuration file
  gslb                         Configure Global SLB features
  gslb-host-policy             GSLB host policy name
  healthck                     Health-check
  http                         HTTP protocol
  ip                           IP settings
  no                           Undo/disable commands
  quit                         Exit to User level
  rshow                        Remote show system information
  server                       Set SLB features
  session                      Set session parameters
  show                         Show system inform

  write                        Write running configuration to flash or terminal
```

If you enter an invalid command while in **configure sync-terminal** or if the command you entered cannot be accepted by the destination ServerIron ADX, a message appears.

Example

```
Peer message: real server 1 not found
```

Also, some commands, such as **no server real**, may take a few seconds to process before the CLI is available for the next command.

To see if the configuration entered at the **configure sync-terminal** level was successfully duplicated on the destination ServerIron ADX, enter the **show run** command on the destination ServerIron ADX. The duplicated configuration should be displayed.

Block-by-block synchronization

This feature allows you to synchronize sections (blocks) of an ServerIron ADX's configuration across a network.

Synchronizing sections of a ServerIron ADX's configuration is useful if you want to synchronize only a portion of the ServerIron ADX's configuration to a peer, or if you want the synchronization to occur manually instead of automatically.

The following sections of the ServerIron ADX's configuration can be synchronized individually.

Synchronizing real server configuration

To synchronize the ServerIron ADX's real server configuration, enter the following commands.

```
ServerIronADX# configure terminal
ServerIronADX(config)# config-sync real-server all
This may remove some configuration on the peer box.
Are you sure? (enter 'y' or 'n'): y
```

Syntax: `config-sync real-server <server-name> | all`

The **config-sync real-server** command synchronizes the device's real server configuration with the peer, but the binding of the real servers to the virtual servers is not retained. Also note that the **sync real-server all** command first removes the existing real server configuration on the peer before applying the new configuration.

Synchronizing virtual server configuration

To synchronize the ServerIron ADX's virtual server configuration, enter the following commands.

```
ServerIronADX# configure terminal
ServerIronADX(config)# config-sync vip all
```

Syntax: `config-sync vip <server-name> | all`

The **config-sync vip** command synchronizes the device's virtual server configuration with the peer, but the binding of the real servers to the virtual servers is not retained. If you are synchronizing the configuration of an individual virtual server, you should synchronize the configurations of the real servers bound to the virtual server, then synchronize the configuration of the virtual server itself. Also note that the **config-sync vip all** command first removes the existing virtual server configuration on the peer before applying the new configuration.

Synchronizing all SLB configurations

To synchronize all of the ServerIron ADX's SLB-related configuration, including real server, virtual server, and URL map configuration, enter the following commands.

```
ServerIronADX# configure terminal
ServerIronADX(config)# config-sync slb
```

Syntax: `config-sync slb`

The **config-sync slb** command synchronizes all the real servers/virtual servers and all the URL maps with the peer and maintains the binding relationship between the real servers and virtual servers, as well as URL maps and virtual servers. Also note that the **config-sync slb** command first removes the existing SLB configuration on the peer before applying the new configuration.

Synchronizing port-profile configuration

To synchronize the ServerIron ADX's port profile configuration, enter the following commands.

1 Displaying system information

```
ServerIronADXA# configure terminal
ServerIronADXA(config)# config-sync port-profile all
This may remove some configuration on the peer box.
Are you sure? (enter 'y' or 'n'): y
```

Syntax: `config-sync port-profile <port-number>|all`

The **config-sync real-server** command synchronizes the device's port profile configuration with the peer. The command first removes the existing port profiles on the peer before applying the new configuration.

Synchronizing all of the content switching policy and rule configurations

To synchronize all of the ServerIron ADX's content switching policy and rule configurations, enter commands such as the following.

```
ServerIronADXA# configure terminal
ServerIronADXA(config)# config-sync csw all
This will first remove all the csw policies/rules on the peer box if already
exists.
Are you sure? (enter 'y' or 'n'):
```

Syntax: `config-sync csw all`

The **config-sync csw all** command synchronizes all of the device's csw policies and rules configuration with the peer; however, the association of the policies and rules with the virtual server is not retained. Also note that the **config-sync csw all** command first removes the existing content switching policies and rules on the peer before it applies the new configuration.

NOTE

The **config-sync real | vip | slb** commands will first delete the corresponding real or virtual servers on the peer. If the real or virtual servers on the peer are handling traffic, the deletion may fail, which would prevent the new real or virtual servers from being created. Consequently, you should ensure there are no sessions on the corresponding real or virtual servers on the peer prior to issuing these commands.

Displaying system information

To view the software and hardware details for the system, enter the following command.

```

ServerIronADX# show version
Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Boot Version 02.00.09 Apr 27 2009 17:13:05 PDT label: dobv2
Monitor Version 02.00.09 Apr 27 2009 17:13:05 PDT label: dobv2
System Version 12.00.00 May 1 2009 13:01:28 PDT label: ASM12000dev
AXP Version: 0.00 Dated: 2009/03/31 11:53:57
PAX Version: 0.0 Dated: 2009/01/23 11:46:57
MBRIDGE Version: 0009, Device ID # bebe
Backplane: ServerIronADX 8000, Serial #: 123451İÿÿ
Chassis: ServerIronADX 8000, Serial #: Not-Present
=====
SL slot-mp1: ServerIron Management Mod, ACTIVE
    Serial #: Not-Present
    Part #: Not-Present
=====
SL slot-sf1: ServerIron Switch Fabric Mod
    Serial #: Not-Present
    Part #: Not-Present
    Version #: 111d8037-00-111d802d-0d-01b720
=====
SL slot-sf2: ServerIron Switch Fabric Mod
    Serial #: Not-Present
    Part #: Not-Present
    Version #: 111d8037-00-111d802d-0d-01b720
=====
SL slot-asml: ServerIron 8BP App Switch Mod
    Serial #: Not-Present
    Part #: Not-Present
    Version #: 111d8037-00
    Application Processors: 8
    1333 MHz Power PC processor (version 00008021/0030) 533 MHz bus
    Boot Version 02.00.09 Apr 27 2009 17:12:28 PDT label: dobv2
=====
Active management module:
    1499 MHz Power PC processor (version 00008021/0030) 599 MHz bus
    1408 KB Boot flash
    65536 KB Code flash
    4096 MB DRAM
The system uptime is 3 minutes 6 seconds
The system started at 11:10:36, GMT+00, Tue May 05 2009

The system - boot source: primary, mode: warm start, soft reset, total resets: 0

```

Syntax: show version

Displaying memory information

To display total and available memory, enter the following command.

```

ServerIronADX# show memory
Total Memory:          2097152 K-bytes
Available Memory:      1076444 K-bytes
BP1: total 2097152 K-bytes, available 411120 K-bytes
BP2: total 2097152 K-bytes, available 411396 K-bytes
BP3: total 2097152 K-bytes, available 411396 K-bytes
BP4: total 2097152 K-bytes, available 411396 K-bytes

```

Syntax: show memory

Displaying chassis information

To display chassis information, enter the following command.

```
ServerIronADX8000# show chassis
Boot Prom MAC: 001b.ed06.0620
Type:      SI-1216-4-PREM
          Serial #:   E20548F4EH
=====
Fan and Power Supply Status
=====
Fan 1 - STATUS: OK SPEED: HI(RPM 9410)
Fan 2 - STATUS: OK SPEED: HI(RPM 9247)
Fan 3 - STATUS: OK SPEED: HI(RPM 9259)
Fan 4 - STATUS: OK SPEED: HI(RPM 9410)
Fan 5 - STATUS: OK SPEED: HI(RPM 9554)
Fan 6 - STATUS: OK SPEED: HI(RPM 9398)
Power 1: not present
Power 2 (Part#: 3I50      Serial#:P119100125   Brocade Part#:32034001) (AC 504W)
          Firmware Ver#: 41. 0: Installed (OK)
Total power budget for system = 504 W
=====
Temperatures per Module
=====
Management CPU Temp: 59 deg C
BP 1 Temp: 64 deg C
BP 2 Temp: 48 deg C
AXP Temp: 35 deg C
```

Displaying module information

To display module information, enter the following command.

```
ServerIronADX8000# show module
```

Slot	Module	Status	Ports	Starting MAC
slot-lc1:	Line Card 12x1G Copper	RUNNING	12	2122.2324.0001
slot-lc2:				
slot-lc3:				
slot-lc4:				
slot-mp1:	ServerIron Management Mod	ACTIVE	0	
slot-mp2:				
slot-sf1:	ServerIron Switch Fabric Mod	RUNNING	0	
slot-sf2:	ServerIron Switch Fabric Mod	RUNNING	0	
slot-asml:	ServerIron 8BP App Switch Mod	RUNNING	0	

Syntax: show module

To display Application Switch module information, enter the following command.

```
ServerIronADX8000# show asm-state
```

slot-asm	Yes	OK	RUNNING
bp 1:	BP App Ready		
bp 2:	BP App Ready		
bp 3:	BP App Ready		
bp 4:	BP App Ready		

Syntax: show asm-state

Displaying and saving tech support information

Commands are provided on the ServerIron ADX that help you display and save information that can help Brocade Technical support troubleshoot your system. These commands are described in the following sections.

Displaying tech support information

To display technical support information use the following command.

```
ServerIronADX# show short-tech-support
set comm-list std_3 delete
!
route-map map1 permit 10
  match ipv6 address prefix-list ipv6_uni
!
route-map test permit 4
  match protocol isis level-1
route-map test permit 99
  set as-path prepend 7701000
!
route-map RMap1 deny 1
  match ipv6 address prefix-list PrefList1
route-map RMap1 permit 2
!
route-map HopMap permit 1
  match ip next-hop 2
!
route-map NetMap permit 1
  match ip address 1
!
route-map TAG_IP permit 1
  match ip address prefix-list p11
  set tag 100
!
route-map static permit 1
  match ip address 10
!
route-map CommMap permit 1
  match community 1
!
route-map GET_ONE permit 1
!
route-map GET_One permit 1
  set as-path prepend 65535
!
route-map Get_One permit 1
!
route-map PathMap permit 1
  match as-path 1
!
route-map set_net permit 1
  set community no-export
!
route-map next-hop permit 10
  match ipv6 address prefix-list next-hop-ipv6
  set ipv6 next-hop 2011:e0ff:3764::34
!
route-map RouteMap1 permit 1
```

1 Displaying system information

```
!
route-map blockuser permit 10
  match tag 50
  set ip next-hop 192.168.0.1
  set local-preference 1000000
  set origin igp
!
route-map test-route permit 99
!
route-map policygroup3 permit 10
  match protocol bgp static-network
  set local-preference 150
!
route-map default_level1 permit 1
  set level level-1
!
route-map dampening_map_enable permit 1
!
route-map dampening_map_neighbor_a permit 1
  set dampening
!
---- BEGIN SSH2 PUBLIC KEY ----
Comment: RomCliSecure DSA Public key
AAAAB3NzaC1kc3MAAACBAKZf6qtRHGHjPfOP3drwO1m28l4fpN5X5c8ArkeKhV3a
TzY404uwCsSvfYQUw/s24E+989MWZxLU00Ib+nV+hWlK0nxI85bQPivOjaWNTbgg
OfNdZ4VyNcLxxzsiJqNhQpGQ3LW2zQ7fsP9pM5ALAs7MDOaSdNja58aUgEMY1ta5
AAAAFQC1r9L5Mkax780fOnwkDB6eIaNjCwAAAH97vSxdyRel4IucL4Ckn7Y/zVwF
eLpwHiVP4lMN7dO2aApuWvsygLU/FUAouv/3PRug/bAAS56w2/JLKVvyolaRPNHA
vgPFEDodqLc+dnC1bXFu1VR69ntQYTEe6iReLlwzeEPLwTW5ucGHddXVbP2jG3R+
JEmGGt87P3JxicCjAAAAGCzJaJ+OTZurGcd6pX96j1+YBxMftUq1Jf43N9AkDDmA
37P4dQW5MFXXvZqlORur8UiBn67scBA4PzHwf0/PK6RH3722ZNQf2F3r6nb571Eh
c9o0lAmwJPsmEXEOShgE5O7u5pelZ86YbDTFuM0a0x25cIM4Px0gnwIsF/ui6lnS
---- END SSH2 PUBLIC KEY ----
!
ip ssh key-authentication no
!
end

Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Boot Version 12.1.00aT405 Jul  9 2010 19:03:54 PDT label: dob12100a
Monitor Version 12.1.00aT405 Jul  9 2010 19:03:54 PDT label: dob12100a
System Version 12.3.01T403 May 26 2011 15:53:29 PDT label: ASR12301
AXP Version: 1.14 Dated: 2010/08/06 10:35:46
PAX Version: 8.4 Dated: 2010/10/06 14:31:02
MBRIDGE Version: 000b, Device ID # bebe

=====
Type:      ServerIron ADX 1216-4-PREM
Backplane Serial #:  E20548F4BK
Chassis Serial #:    E30549F00K
LID #: gIFKJOHFFm License: SI-1216-4-SSL-PREM
Part #: 35723-001
SSL Card - Serial #: 1F0548F090          Part #: 40-1000373-02
Version #: 13b626-03030303-111d8036-00
=====
Active management module:
  1499 MHz Power PC processor (SVR 1.1, PVR 8021/0030) 599 MHz bus
  512 KB Boot flash
  131072 KB Code flash
  2048 MB DRAM
```

The system uptime is 15 days 1 hours 34 minutes 17 seconds
 The system started at 15:38:46, GMT+10 Daylight Time, Mon Aug 29 2011

The system - boot source: secondary, mode: cold start, soft reset, total resets:0

Port	Link	State	Dupl	Speed	Trunk	Tag	Priori	MAC	Name
1	Down	None	None	None	1	No	level0	001b.ed06.0300	marsha markey
2	Down	None	None	None	1	No	level0	001b.ed06.0300	
3	Down	None	None	None	1	No	level0	001b.ed06.0300	
4	Down	None	None	None	1	No	level0	001b.ed06.0300	
5	Down	None	None	None	None	No	level0	001b.ed06.0300	
6	Down	None	None	None	None	No	level0	001b.ed06.0300	
7	Down	None	None	None	None	No	level0	001b.ed06.0306	
8	Down	None	None	None	None	No	level0	001b.ed06.0307	
9	Down	None	None	None	9	No	level0	001b.ed06.0300	
10	Up	Forward	Full	1G	9	No	level0	001b.ed06.0300	
11	Down	None	None	None	None	No	level0	001b.ed06.0300	
12	Down	None	None	None	None	No	level0	001b.ed06.0300	
13	Down	None	None	None	None	No	level0	001b.ed06.030c	
14	Down	None	None	None	None	No	level0	001b.ed06.0300	
15	Down	None	None	None	None	No	level0	001b.ed06.0300	
16	Down	None	None	None	None	No	level0	001b.ed06.030f	
17	Down	None	None	None	None	No	level0	001b.ed06.0310	
18	Down	None	None	None	None	No	level0	001b.ed06.0311	
mgmt1	UP	Forward	FULL	1G	None	No	level0	001b.ed06.0300	
ve1	Down	N/A	N/A	N/A	None	N/A	N/A	001b.ed06.0300	
ve4	Down	N/A	N/A	N/A	None	N/A	N/A	001b.ed06.0300	
ve5	Up	N/A	N/A	N/A	None	N/A	N/A	001b.ed06.0300	
lb1	Down	N/A	N/A	N/A	None	N/A	N/A	N/A	
lb2	Up	N/A	N/A	N/A	None	N/A	N/A	N/A	

Buffer Manager Queue

[Pkt Receive Pkt Transmit]

0 0

Ethernet Port	Packets		Collisions		Errors	
	[Receive	Transmit]	[Recv	Txmit]	[InErr	OutErr]
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	4297598	655976	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
mgmt1	87439500	2546	0	0	0	0

Syslog logging: enabled (0 messages dropped, 0 flushes, 29 overruns)

Buffer logging: level ACDMEINW, 50 messages logged

level code: A=alert C=critical D=debugging M=emergency E=error

I=informational N=notification W=warning

1 Displaying system information

Static Log Buffer:

Aug 29 15:39:24:A:Power supply 1, 1st from left, (NOT PRESENT)

Aug 29 15:46:07:A:BP1&2 Temperature 82 C degrees is normal

Dynamic Log Buffer (50 lines):

```
Sep 13 17:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 16:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 16:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 16:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 16:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 15:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 15:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 15:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 15:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 14:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 14:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 14:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 14:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 13:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 13:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 13:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 13:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 12:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 12:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 12:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 12:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 11:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 11:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 11:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 11:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 10:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 10:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 10:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 10:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 09:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 09:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 09:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 09:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 08:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 08:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 08:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 08:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 07:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 07:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 07:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 07:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 06:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 06:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 06:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 06:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 05:54:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 05:39:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 05:24:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 05:09:42:W:NTP server 208.99.8.95 failed to respond
Sep 13 04:54:42:W:NTP server 208.99.8.95 failed to respond
```

Syntax: show short-tech-support

Saving tech support information to a file

You can save detailed technical information to a file to the internal USB drive of the ServerIron ADX for assistance in troubleshooting issues when working with technical support.

```
ServerIronADX1000# save tech-support text test1
Msg: tech-support info to be saved in test1
Retrieving save tech infomation, please wait...
```

```
checking bp dumps on usb0
start to write file to flash.....Done saving tech-support info to file.
```

Type cntrl Y and then m on the console to go into the OS mode.

```
OS> dir
2093704 [0000] A1B12000.bin
6823553 [0000] A1B12100.bin
```

```
...
454089 [4bca] test1
50439658 bytes 14 File(s)
80216064 bytes free
```

```
USB 0 drive:
Directory is empty
```

```
4102352896 bytes
4102348800 bytes free
```

Syntax: `save tech-support text | html <file-name>`

The **text** parameter specifies that the technical support information be saved as a plain text file.

The **html** parameter specifies that the technical support information be saved in HTML format.

The `<file-name>` variable specifies the name of the file that the technical support information will be saved to.

NOTE

A typical output file is greater than 10 MB in size and can be much larger if there are many crash dumps (either MPs or BPs).

Displaying statistics

To display statistics, enter a command such as the following.

1 Displaying system information

```
ServerIronADX# show statistics brief
Buffer Manager      Queue
[Pkt Receive Pkt Transmit]
          0          0
```

Ethernet Port	Packets		Collisions		Errors	
	[Receive	Transmit]	[Recv	Txmit]	[InErr	OutErr]
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	615934	4033121	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0

```
ServerIronADX#
```

Syntax: `show statistics ethernet <portnum> | slot <slot-num> | pos <pos-port> | brief | dos-attack`

The **pos** <portnum> parameter displays statistics for a specific POS port.

The **ethernet** <portnum> parameter displays statistics for a specific Ethernet port.

The **slot** <slot-num> parameter displays statistics for a specific chassis slot.

The display shows the following information for each port.

TABLE 3 CLI display of port statistics

This field...	Displays...
Packet counters	
Receive	The number of packets received on this interface.
Transmit	The number of packets transmitted on this interface.
Collision counters	
Receive	The number of collisions that have occurred when receiving packets.
Transmit	The number of collisions that have occurred when sending packets.
Packet Errors	
These fields show statistics for various types of packet errors. The device drops packets that contain one of these errors.	
Align	The number of packets that contained frame alignment errors.
FCS	The number of packets that contained Frame Check Sequence errors.
Giant	The number of packets that were longer than the configured MTU.
Short	The number of packets that were shorter than the minimum valid length.

Displaying port statistics

Port statistics are polled by default every 10 seconds.

You can view statistics for ports by entering the following **show** commands:

- show interfaces
- show configuration

Displaying STP statistics

You can view a summary of STP statistics on the ServerIron ADX. STP statistics are by default polled every 10 seconds.

To view spanning tree statistics, enter the show span command. To view STP statistics for a VLAN, enter the **span vlan** command.

Displaying trunk group information

To display trunk groups and their port membership for ServerIron ADXs, enter the following command.

```
ServerIronADX# Show trunk
Configured trunks:
Trunk ID: 1
Type: Server
Ports_Configured: 2
Primary Port Monitored: Jointly

Ports          1      2
Port Names     none   none
Port_Status    enable enable
Monitor        off    off
Mirror Port    N/A    N/A
Monitor Dir    N/A    N/A

Trunk ID: 3
Type: Switch
Ports_Configured: 2
Primary Port Monitored: Individually

Ports          3      4
Port Names     custom* none
Port_Status    enable enable
```

Clearing the statistics

To globally clear all counters on the system, enter the following command.

```
ServerIronADX# clear statistics ?
dos-attack      Clear DOS-attack statistics
ethernet        Ethernet port
pos             POS port
rate-counters
slot            Module slot
<cr>
```

Syntax: `clear statistics [<options>]`

Clearing all sessions

In rare instances, it may be necessary to delete all the sessions on the ServerIron ADX at once. You can delete all regular (non-static) sessions on the ServerIron ADX, by entering the following command (Use this command with caution).

```
ServerIronADX# clear server all-session
```

When you enter this command, all regular (non-static) sessions on the ServerIron ADX are deleted. The command removes both active sessions as well as stale sessions in the delete queue.

Syntax: `clear server all-session`

Using Syslog

The ServerIronADX contains a syslog agent that writes log messages to a local buffer and optionally to a third-party syslog server. The syslog feature can write messages at the following severity levels.

The device automatically writes the syslog messages to a local buffer. If you specify the IP address or name of a syslog server, the device also writes the messages to the syslog server. The default facility for messages written to the server is "user". You can change the facility if needed. You also can change the number of entries that can be stored in the local buffer. The default is 50. The ServerIron ADX does not have a limit to the number of messages that can be logged on a remote syslog server.

NOTE

You can specify only one facility.

Severity levels

The syslog agent writes messages to provide information about the following severity levels:

- Emergencies
- Alerts
- Critical
- Errors
- Warnings
- Notifications
- Informational
- Debugging

The device writes the messages to a local buffer that can hold up to 100 messages. You also can specify the IP address or host name of up to six syslog servers. When you specify a syslog server, the Brocade device writes the messages both to the system log and to the syslog server.

Using a syslog server ensures that the messages remain available even after a system reload. The Brocade device's local syslog buffer is cleared during a system reload or reboot, but the syslog messages sent to the syslog server remain on the server.

The syslog service on a syslog server receives logging messages from applications on the local host or from devices such as a router or switch. Syslog adds a time stamp to each received message and directs messages to a log file. Most Unix workstations come with syslog configured. Some third party vendor products also provide syslog running on NT.

Syslog uses UDP port 514 and each syslog message thus is sent with destination port 514. Each syslog message is one line with syslog message format. The message is embedded in the text portion of the syslog format. There are several subfields in the format. Keywords are used to identify each subfield, and commas are delimiters. The subfield order is insensitive except that the text subfield should be the last field in the message. All the subfields are optional.

Configuring logging

You can enable or disable logging, configure the size of the local log buffer, and specify a Syslog server, by entering the **logging** command.

To disable logging of SNMP traps to a locally saved event log, enter the following command.

```
ServerIronADX(config)# no logging on
```

To re-enable logging, enter the following command.

```
ServerIronADX(config)# logging on
```

By default, a message is logged whenever a user logs into or out of the CLI's User EXEC or Privileged EXEC mode. To disable logging of users' CLI access, enter the following command.

```
ServerIronADX(config)#no logging enable user-login
```

To specify two third-party Syslog servers to receive Syslog messages in addition to the device's local Syslog buffer, enter commands such as the following.

```
ServerIronADX(config)# logging 10.0.0.99
ServerIronADX(config)# logging 209.157.23.69
```

To change the logging facility from the default facility **user** to **local7**, enter the following command.

```
ServerIronADX(config)#logging facility local7
```

To disable logging of debugging and informational messages, enter commands such as the following.

```
ServerIronADX(config)#no logging buffered debugging
ServerIronADX(config)#no logging buffered informational
```

Syntax: **[no] logging on | enable | <ip-addr> | facility <value> | buffered <level> | console**

The **<level>** parameter can be alerts, critical, debugging, emergencies, errors, informational, notifications, or warnings. All message levels are enabled by default. You can disable message levels individually. The **<num-entries>** can be 1 – 100. All message levels are logged by default. The default local buffer capacity is 50 entries

Possible facility values include:

- **kern** – kernel messages
- **user** – random user-level messages (default)
- **mail** – mail system
- **daemon** – system daemons
- **auth** – security or authorization messages
- **syslog** – messages generated internally by Syslog
- **lpr** – line printer subsystem
- **news** – netnews subsystem
- **uucp** – uucp subsystem
- **sys9** – cron or at subsystem
- **sys10** – reserved for system use
- **sys11** – reserved for system use
- **sys12** – reserved for system use
- **sys13** – reserved for system use
- **sys14** – reserved for system use
- **cron** – cron or at subsystem
- **local0** – reserved for local use
- **local1** – reserved for local use
- **local2** – reserved for local use
- **local3** – reserved for local use
- **local4** – reserved for local use
- **local5** – reserved for local use
- **local6** – reserved for local use
- **local7** – reserved for local use

Displaying log information

To display the syslog messages in the device's local log file, enter the following command.

```
ServerIronADX# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 33 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
00d01h02m36s:I:Interface   ethernet2/19, state down
00d01h02m36s:I:VLAN 1 Port 2/19 STP State -> DISABLED (PortDown)
00d00h39m55s:I:VLAN 1 Port 2/24 STP State -> FORWARDING (FwdDlyExpiry)
00d00h39m53s:I:VLAN 1 Port 2/24 STP State -> LEARNING (FwdDlyExpiry)
00d00h39m51s:I:Interface   ethernet2/24, state up
00d00h39m51s:I:VLAN 1 Port 2/24 STP State -> LISTENING (MakeFwding)
00d00h36m49s:I:VLAN 1 Port 2/19 STP State -> FORWARDING (FwdDlyExpiry)
00d00h36m47s:I:VLAN 1 Port 2/19 STP State -> LEARNING (FwdDlyExpiry)
00d00h36m45s:I:Interface   ethernet2/19, state up
00d00h36m45s:I:VLAN 1 Port 2/19 STP State -> LISTENING (MakeFwding)
```

This example shows log entries for authentication failures. If someone enters an invalid community string when attempting to access the SNMP server on the Brocade device, the device generates a trap in the device's Syslog buffer. (If you have configured the device to use a third-party Syslog server, the device also sends a log entry to the server.)

Syntax: show logging

Here is an example of a log that contains SNMP authentication traps. In this example, someone attempted to access the Brocade device three times using invalid SNMP community strings. The unsuccessful attempts indicate either an authorized user who is also a poor typist, or an unauthorized user who is attempting to access the device.

```
ServerIronADX(config)# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 1 overruns)
Buffer logging: level ACDMEINW, 50 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning
```

Log Buffer (50 entries):

```
00d01h45m13s:info:SNMP Authentication failure, intruder IP: 207.95.6.55
00d00h01m00s:info:SNMP Authentication failure, intruder IP: 207.95.6.55
00d00h00m05s:info:SNMP Authentication failure, intruder IP: 207.95.6.55
```

This example shows a log entry for an IP address conflict between the Brocade device and another device on the network.

In addition to placing an entry in the log, the software sends a log message to the Syslog server, if you have configured one, and sends a message to each open CLI session.

```
ServerIronADX(config)# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 1 overruns)
Buffer logging: level ACDMEINW, 50 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning
```

Log Buffer (50 entries):

```
00d01h45m13s:warning:Duplicate IP address 209.157.23.188 detected, sent from MAC
address 00e0.5201.3bc9 coming from port 7/7
```

Here are some examples of log entries for packets denied by Access Control Lists (ACLs).

NOTE

On devices that also use Layer 2 MAC filters, both types of log entries can appear in the same log. Only ACL log entries are shown in this example.

1 Using Syslog

```
ServerIronADX(config)# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
  Buffer logging: level ACDMEINW, 38 messages logged
  level code: A=alert C=critical D=debugging M=emergency E=error
              I=informational N=notification W=warning

Log Buffer (50 entries):

21d07h02m40s:warning:list 101 denied tcp 209.157.22.191(0)(Ethernet 4/18
0010.5alf.77ed) -> 198.99.4.69(http), 2 packets
00d07h03m30s:warning:list 101 denied tcp 209.157.22.26(0)(Ethernet 4/18
0010.5alf.77ed) -> 198.99.4.69(http), 2 packets
00d06h58m30s:warning:list 101 denied tcp 209.157.22.198(0)(Ethernet 4/18
0010.5alf.77ed) -> 198.99.4.69(http), 1 packets
```

The first time an entry in an ACL denies a packet and logging is enabled for that entry, the software generates a Syslog message and an SNMP trap. Messages for packets denied by ACLs are at the warning level of the Syslog.

When the first Syslog entry for a packet denied by an ACL is generated, the software starts a five-minute ACL timer. After this, the software sends Syslog messages every five minutes. The messages list the number of packets denied by each ACL during the previous five-minute interval. If an ACL entry does not deny any packets during the five-minute interval, the software does not generate a Syslog entry for that ACL entry.

NOTE

For an ACL entry to be eligible to generate a Syslog entry for denied packets, logging must be enabled for the entry. The Syslog contains entries only for the ACL entries that deny packets and have logging enabled.

In this example, the two-line message at the bottom is the first entry, which the software immediately generates the first time an ACL entry permits or denies a packet. In this case, an entry in ACL 101 denied a packet. The packet was a TCP packet from host 209.157.22.198 and was destined for TCP port 80 (HTTP) on host 198.99.4.69.

When the software places the first entry in the log, the software also starts the five-minute timer for subsequent log entries. Thus, five minutes after the first log entry, the software generates another log entry and SNMP trap for denied packets.

In this example, the software generates the second log entry five minutes later. The second entry indicates that the same ACL denied two packets.

The time stamp for the third entry is much later than the time stamps for the first two entries. In this case, no ACLs denied packets for a very long time. In fact, since no ACLs denied packets during the five-minute interval following the second entry, the software stopped the ACL log timer. The software generated the third entry as soon as the ACL denied a packet. The software restarted the five-minute ACL log timer at the same time. As long as at least one ACL entry permits or denies a packet, the timer continues to generate new log entries and SNMP traps every five minutes.

Here are some examples of log messages for CLI access.

```

ServerIronADX(config)# show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
Buffer logging: level ACDMEINW, 12 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning
Log Buffer (50 entries):
Oct 15 18:01:11:info:dg logout from USER EXEC mode
Oct 15 17:59:22:info:dg logout from PRIVILEGE EXEC mode
Oct 15 17:38:07:info:dg login to PRIVILEGE EXEC mode
Oct 15 17:38:03:info:dg login to USER EXEC mode

```

The first message (the one on the bottom) indicates that user “dg” logged in to the CLI’s User EXEC level on October 15 at 5:38 PM and 3 seconds (Oct 15 17:38:03). The same user logged in to the Privileged EXEC level four seconds later.

The user remained in the Privileged EXEC mode until 5:59 PM and 22 seconds. (The user could have used the CONFIG modes as well. Once you access the Privileged EXEC level, no further authentication is required to access the CONFIG levels.) At 6:01 PM and 11 seconds, the user ended the CLI session.

Clearing syslog entries

To remove the syslog entries displayed by the **show logging** command, enter the following command.

```
ServerIronADX# clear logging
```

Syntax: clear logging

Message format

[Table 4](#) lists the general format and explanation of a syslog message at the following message levels:

- Emergencies (none)
- Alerts
- Critical (none)
- Errors (none)
- Warnings
- Notifications
- Informational
- Debugging

TABLE 4 Brocade Syslog messages

Message Level	Message Format	Explanation
Alert	Power supply <num>, <location>, failed	<p>A power supply has failed.</p> <p>The <num> is the power supply number.</p> <p>The <location> describes where the failed power supply is in the chassis. The location can be one of the following:</p> <p>ServerIron ADX 8000:</p> <ul style="list-style-type: none"> • Left Most • Second from Left • Third from Left • Last <p>ServerIron ADX 4000:</p> <ul style="list-style-type: none"> • left • right <p>ServerIron ADX 1000:</p> <ul style="list-style-type: none"> • left • right <p>ServerIron ADX 10000:</p> <ul style="list-style-type: none"> • top • second from top • third from top • bottom
Alert	Fan <num>, <location>, failed	<p>A fan has failed.</p> <p>The <num> is the power supply number.</p> <p>The <location> describes where the failed power supply is in the chassis. The location can be one of the following:</p> <p>ServerIron ADX 8000:</p> <ul style="list-style-type: none"> • Back top • Front top • Back Middle • Front Middle • Back bottom • Front bottom <p>ServerIron ADX 4000:</p> <ul style="list-style-type: none"> • Rear • Middle • Front <p>ServerIron ADX 1000:</p> <ul style="list-style-type: none"> • Fans 1 to 6 (numbered left-to-right – All in the same controller) <p>ServerIron ADX 10000:</p> <ul style="list-style-type: none"> • Left most • Second from left • Third from left • Last

TABLE 4 Brocade Syslog messages (Continued)

Message Level	Message Format	Explanation
Alert	Management module at slot <i><slot-num></i> state changed from <i><module-state></i> to <i><module-state></i> .	<p>Indicates a state change in a management module. The <i><slot-num></i> indicates the chassis slot containing the module.</p> <p>The <i><module-state></i> can be one of the following:</p> <ul style="list-style-type: none"> • active • standby • crashed • coming-up • unknown
Alert	Temperature <i><module></i> <i><degrees></i> C degrees, warning level <i><warn-degrees></i> C degrees, shutdown level <i><shutdown-degrees></i> C degrees	<p>Indicates an overtemperature condition on the module specified by the <i><module></i> variable.</p> <p>The <i><degrees></i> value indicates the temperature of the module.</p> <p>The <i><warn-degrees></i> value is the warning threshold temperature configured for the module.</p> <p>The <i><shutdown-degrees></i> value is the shutdown temperature configured for the module.</p>
Alert	<i><num-modules></i> modules and 1 power supply, need more power supply!!	<p>Indicates that the Chassis device needs more power supplies to run the modules in the chassis.</p> <p>The <i><num-modules></i> parameter indicates the number of modules in the chassis.</p>
Alert	Out of tcp send buffer at <i><application></i>	<p>Indicates that the TCP send buffer is exhausted.</p> <p>The <i><application></i> parameter is the application that caused the buffer overflow.</p>
Alert	Out of TCB memory at <i><application></i>	<p>Indicates that TCB memory is exhausted.</p> <p>The <i><application></i> parameter shows which application is out of TCB memory.</p>
Warning	Locked address violation at interface e <i><portnum></i> , address <i><mac-address></i>	<p>Indicates that a port on which you have configured a lock-address filter received a packet that was dropped because the packet's source MAC address did not match an address learned by the port before the lock took effect.</p> <p>The e<i><portnum></i> is the port number.</p> <p>The <i><mac-address></i> is the MAC address that was denied by the address lock.</p> <p>Assuming that you configured the port to learn only the addresses that have valid access to the port, this message indicates a security violation.</p>
Warning	NTP server <i><ip-addr></i> failed to respond	<p>Indicates that a Simple Network Time Protocol (SNTP) server did not respond to the device's query for the current time.</p> <p>The <i><ip-addr></i> indicates the IP address of the SNTP server.</p>

TABLE 4 Brocade Syslog messages (Continued)

Message Level	Message Format	Explanation
Warning	Dup IP <i><ip-addr></i> detected, sent from MAC <i><mac-addr></i> interface <i><portnum></i>	Indicates that the Brocade device received a packet from another device on the network with an IP address that is also configured on the Brocade device. The <i><ip-addr></i> is the duplicate IP address. The <i><mac-addr></i> is the MAC address of the device with the duplicate IP address. The <i><portnum></i> is the Brocade port that received the packet with the duplicate IP address. The address is the packet's source IP address.
Warning	mac filter group denied packets on port <i><portnum></i> src macaddr <i><mac-addr></i> , <i><num></i> packets	Indicates that a Layer 2 MAC filter group configured on a port has denied packets. The <i><portnum></i> is the port on which the packets were denied. The <i><mac-addr></i> is eth source AMC address of the denied packets. The <i><num></i> indicates how many packets matching the values above were dropped during the five-minute interval represented by the log entry.
Warning	list <i><acl-num></i> denied <i><ip-proto></i> <i><src-ip-addr></i> (<i><src-tcp/udp-port></i>) (Ethernet <i><portnum></i> <i><mac-addr></i>) -> <i><dst-ip-addr></i> (<i><dst-tcp/udp-port></i>), 1 event(s)	Indicates that an Access Control List (ACL) denied (dropped) packets. The <i><acl-num></i> indicates the ACL number. Numbers 1 – 99 indicate standard ACLs. Numbers 100 – 199 indicate extended ACLs. The <i><ip-proto></i> indicates the IP protocol of the denied packets. The <i><src-ip-addr></i> is the source IP address of the denied packets. the <i><src-TCP/UDP-port></i> is the source TCP or UDP port, if applicable, of the denied packets. The <i><portnum></i> indicates the port number on which the packet was denied. The <i><mac-addr></i> indicates the source MAC address of the denied packets. The <i><dst-ip-addr></i> indicates the destination IP address of the denied packets. The <i><dst-tcp/udp-port></i> indicates the destination TCP or UDP port number, if applicable, of the denied packets.
Warning	firewall group <i><groupnum></i> become active	Indicates that this ServerIron ADX has become the active ServerIron ADX in the high-availability (active-standby) FWLB configuration. (High-availability FWLB configurations also are called "IronClad" configurations.) The <i><groupnum></i> is the FWLB group ID, which normally is 2.
Warning	firewall group <i><groupnum></i> become standby	Indicates that this ServerIron ADX has become the standby ServerIron ADX in the high-availability (active-standby) FWLB configuration. (High-availability FWLB configurations also are called "IronClad" configurations.) The <i><groupnum></i> is the FWLB group ID, which normally is 2.

TABLE 4 Brocade Syslog messages (Continued)

Message Level	Message Format	Explanation
Warning	firewall path up target <i><target-ip-addr></i> nexthop <i><next-hop-ip-addr></i> path <i><path-id></i> port <i><portnum></i>	<p>Indicates that a firewall path has come up (and is therefore good).</p> <p>The <i><target-ip-addr></i> is the IP interface at the remote end of the path.</p> <p>The <i><next-hop-ip-addr></i> is the IP interface of the next hop in the path.</p> <p>The <i><path-id></i> is the ID you assigned to the path when you configured it.</p> <p>The <i><portnum></i> is the ServerIron ADX port connected to the path's next hop.</p>
Warning	firewall path down target <i><target-ip-addr></i> nexthop <i><next-hop-ip-addr></i> path <i><path-id></i> port <i><portnum></i>	<p>Indicates that a firewall path has gone down (and is therefore unusable).</p> <p>The <i><target-ip-addr></i> is the IP interface at the remote end of the path.</p> <p>The <i><next-hop-ip-addr></i> is the IP interface of the next hop in the path.</p> <p>The <i><path-id></i> is the ID you assigned to the path when you configured it.</p> <p>The <i><portnum></i> is the ServerIron ADX port connected to the path's next hop.</p>
Warning	HTTP match-list <i><matching-list></i> with simple pattern <i><string></i> Alert: bring server Down.	<p>Indicates that an HTTP content verification health check has matched a set of selection criteria specified in a down simple statement.</p> <p>When the selection criteria is found in the HTML file used for the health check, the ServerIron ADX marks port 80 (HTTP) on the real server FAILED.</p> <p><i><matching-list></i> is the name of the matching list whose selection criteria was matched.</p> <p><i><string></i> is the selection criteria.</p>
Warning	HTTP match-list <i><policy-name></i> with simple pattern <i><string></i> Alert: bring server Up.	<p>Indicates that an HTTP content verification health check has matched a set of selection criteria specified in a up simple statement.</p> <p>When the selection criteria is found in the HTML file used for the health check, the ServerIron ADX marks port 80 (HTTP) on the real server ACTIVE.</p> <p><i><policy-name></i> is the name of the matching list whose selection criteria was matched.</p> <p><i><string></i> is the selection criteria.</p>
Warning	HTTP match-list <i><matching-list></i> with compound pattern1 <i><start></i> and pattern2 <i><end></i> Alert: bring server down and Extract message: <i><text-between-start-and-end-pattern></i>	<p>Indicates that an HTTP content verification health check has matched a set of selection criteria specified in a down compound statement.</p> <p>When the selection criteria is found in the HTML file used for the health check, the ServerIron ADX marks port 80 (HTTP) on the real server FAILED.</p> <p><i><matching-list></i> is the name of the matching list whose selection criteria was matched.</p> <p><i><start></i> is the beginning of the selection criteria.</p> <p><i><end></i> is the end of the selection criteria.</p>

TABLE 4 Brocade Syslog messages (Continued)

Message Level	Message Format	Explanation
Warning	HTTP match-list <i><matching-list></i> with compound pattern1 <i><start></i> and pattern2 <i><end></i> Alert: bring server up and Extract message: <i><text-between-start-and-end-pattern></i>	Indicates that an HTTP content verification health check has matched a set of selection criteria specified in a up compound statement. When the selection criteria is found in the HTML file used for the health check, the ServerIron ADX marks port 80 (HTTP) on the real server ACTIVE. <i><matching-list></i> is the name of the matching list whose selection criteria was matched. <i><start></i> is the beginning of the selection criteria. <i><end></i> is the end of the selection criteria.
Warning	Port <i><TCP/UDP-portnum></i> on server <i><name></i> : <i><ip-addr></i> : Avg response time <i><num></i> exceeded lower threshold	The application port on the real server did not respond within the warning threshold time. The <i><TCP/UDP-portnum></i> is the application port number. The <i><name></i> is the real server name. The <i><ip-addr></i> is the real server IP address. The <i><num></i> is the average number of milliseconds it was taking the application port to respond.
Warning	Port <i><TCP/UDP-portnum></i> on server <i><name></i> : <i><ip-addr></i> : Avg response time <i><num></i> exceeded upper threshold; Bringing down the port...	The application port on the real server did not respond within the shutdown threshold time. The <i><TCP/UDP-portnum></i> is the application port number. The <i><name></i> is the real server name. The <i><ip-addr></i> is the real server IP address. The <i><num></i> is the average number of milliseconds it was taking the application port to respond.
Notification	Module was inserted to slot <i><slot-num></i>	Indicates that a module was inserted into a chassis slot. The <i><slot-num></i> is the number of the chassis slot into which the module was inserted.
Notification	Module was removed from slot <i><slot-num></i>	Indicates that a module was removed from a chassis slot. The <i><slot-num></i> is the number of the chassis slot from which the module was removed.
Notification	L4 max connections <i><num></i> reached	Indicates that the maximum number of connections supported by the ServerIron ADX has been reached. The <i><num></i> indicates the number of connections.
Notification	L4 TCP SYN limits <i><num></i> reached	Indicates that the maximum number of connections per second allowed by the ServerIron ADX has been reached. The <i><num></i> indicates the number of connections.
Notification	L4 server <i><ip-addr></i> <i><name></i> max connections <i><num></i> reached	Indicates that the maximum number of connections allowed on a real server has been reached. The <i><ip-addr></i> is the real server's IP address. The <i><name></i> is the name of the real server. The <i><num></i> indicates the number of connections.
Notification	L4 begin-holddown source-ip <i><src-ip-addr></i> dest-ip <i><dst-ip-addr></i>	Indicates that the ServerIron ADX's SYN attack prevention feature is "holding down" the specified source and destination IP address pair, which means the ServerIron ADX is not sending these packets to any servers.

TABLE 4 Brocade Syslog messages (Continued)

Message Level	Message Format	Explanation
Notification	L4 server <ip-addr> <name> is up	Indicates that a real server or cache server has come up. The <ip-addr> is the server's IP address. The <name> is the name of the server.
Notification	L4 server <ip-addr> <name> is down due to <reason>	Indicates that a real server or cache server has gone down. The <ip-addr> is the server's IP address. The <name> is the name of the server. The <reason> is the reason the ServerIron ADX changed the port's state to down. The <reason> can be one of the following: <ul style="list-style-type: none"> healthck – The port failed a health check. This applies to standard health checks and Boolean health checks. reassign – The reassign threshold was reached. server-down – The server failed the Layer 3 health check when you bound the real server to the VIP. MAC-delete – The server's MAC address was deleted from the ServerIron ADX MAC table. graceful-shutdown – The server was gracefully shut down. mp-port-state-change – The port was brought down on the BP managing the real server, in response to a message from the MP CPU that the port is down. NOTE: This value applies only to ServerIron ADX Chassis devices. <ul style="list-style-type: none"> other – The port was brought down by another application (by something other than the ServerIron ADX.) unknown – The port was brought down by a reason other than one of those listed above.
Notification	L4 server <ip-addr> <name> TCP port <tcp-port-num> is up	Indicates that a real server's or cache server's TCP port has come up. The <ip-addr> is the server's IP address. The <name> is the name of the server. The <tcp-port-num> is the TCP port number.
Notification	L4 server <ip-addr> <name> TCP port <tcp-port-num> is down	Indicates that a real server's or cache server's TCP port has gone down. The <ip-addr> is the server's IP address. The <name> is the name of the server. The <tcp-port-num> is the TCP port number.
Notification	L4 switch changed state from active to standby	The ServerIron ADX is an active-standby configuration and has changed from the active to the standby state.
Notification	L4 switch changed state from standby to active	The ServerIron ADX is an active-standby configuration and has changed from the standby to the active state.
Notification	L4 gslb connection to site <name> ServerIronADX <ip-addr> <name> is up	The GSLB protocol connection from this GSLB ServerIron ADX to a remote site ServerIron ADX has come up. The first <name> the site name. The <ip-addr> and <name> are the site ServerIron ADX's management IP address and name.

TABLE 4 Brocade Syslog messages (Continued)

Message Level	Message Format	Explanation
Notification	L4 gslb connection to site <name> ServerIronADX <ip-addr> <name> is down	The GSLB protocol connection from this GSLB ServerIron ADX to a remote site ServerIron ADX went down. The first <name> the site name. The <ip-addr> and <name> are the site ServerIron ADX's management IP address and name.
Notification	L4 gslb connection to gslb ServerIronADX <ip-addr> is up	The GSLB protocol connection from this site ServerIron ADX to a remote GSLB ServerIron ADX has come up. The <ip-addr> is the GSLB ServerIron ADX's management IP address.
Notification	L4 gslb connection to gslb ServerIronADX <ip-addr> is down	The GSLB protocol connection from this site ServerIron ADX to a remote GSLB ServerIron ADX has gone down. The <ip-addr> is the GSLB ServerIron ADX's management IP address.
Notification	L4 gslb health-check <ip-addr> of <zone> status changed to up	The IP address belonging to a domain name for which the ServerIron ADX is providing GSLB has come up. The <ip-addr> is the IP address in the DNS reply. The <zone> is the zone name.
Notification	L4 gslb health-check <ip-addr> of <zone> status changed to down	The IP address belonging to a domain name for which the ServerIron ADX is providing GSLB has gone down. The <ip-addr> is the IP address in the DNS reply. The <zone> is the zone name.
Notification	L4 gslb health-check <ip-addr> of <zone> port <tcp/udp-port> is up	An application port in a domain on the site IP address passed its Layer 4 TCP or UDP health check. The <ip-addr> is the IP address in the DNS reply. The <zone> is the zone name. The <tcp/udp-port> is the application port.
Notification	L4 gslb health-check <ip-addr> of <zone> port <tcp/udp-port> is down	An application port in a domain on the site IP address failed its Layer 4 TCP or UDP health check. The <ip-addr> is the IP address in the DNS reply. The <zone> is the zone name. The <tcp/udp-port> is the application port.
Informational	Cold start	The device has been powered on.
Informational	Warm start	The system software (flash code) has been reloaded.
Informational	<user-name> login to USER EXEC mode	A user has logged into the USER EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> logout from USER EXEC mode	A user has logged out of the USER EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> login to PRIVILEGED mode	A user has logged into the Privileged EXEC mode of the CLI. The <user-name> is the user name.
Informational	<user-name> logout from PRIVILEGED mode	A user has logged out of Privileged EXEC mode of the CLI. The <user-name> is the user name.

TABLE 4 Brocade Syslog messages (Continued)

Message Level	Message Format	Explanation
Informational	SNMP Auth. failure, intruder IP: <i><ip-addr></i>	A user has tried to open a management session with the device using an invalid SNMP community string. The <i><ip-addr></i> is the IP address of the host that sent the invalid community string.
Informational	Interface <i><portnum></i> , state up	A port has come up. The <i><portnum></i> is the port number.
Informational	Interface <i><portnum></i> , state down	A port has gone down. The <i><portnum></i> is the port number.
Informational	Bridge root changed, vlan <i><vlan-id></i> , new root ID <i><root-id></i> , root interface <i><portnum></i>	A Spanning Tree Protocol (STP) topology change has occurred. The <i><vlan-id></i> is the ID of the VLAN in which the STP topology change occurred. The <i><root-id></i> is the STP bridge root ID. The <i><portnum></i> is the number of the port connected to the new root bridge.
Informational	Bridge is new root, vlan <i><vlan-id></i> , root ID <i><root-id></i>	A Spanning Tree Protocol (STP) topology change has occurred, resulting in the Brocade device becoming the root bridge. The <i><vlan-id></i> is the ID of the VLAN in which the STP topology change occurred. The <i><root-id></i> is the STP bridge root ID.
Informational	Bridge topology change, vlan <i><vlan-id></i> , interface <i><portnum></i> , changed state to <i><stp-state></i>	A Spanning Tree Protocol (STP) topology change has occurred on a port. The <i><vlan-id></i> is the ID of the VLAN in which the STP topology change occurred. The <i><portnum></i> is the port number. The <i><stp-state></i> is the new STP state and can be one of the following: <ul style="list-style-type: none"> • disabled • blocking • listening • learning • forwarding • unknown

Event logging

The Event Logging feature of the ServerIron ADX captures all of the activity on the MP and BP consoles and saves it in a file named "eventlog.txt" which is saved on the internal USB drive. This log captures all of the following information along with a timestamp:

- Syslog messages when they are logged
- All MP console messages (boot and application)
- All BP console messages on each Core (boot and application)
- All CLI commands typed by users at the MP and BP consoles
- All commands typed at the OS prompt

Disabling and enabling event logging

The event logging feature is enabled by default. You can disable event logging as shown in the following.

```
ServerIronADX(config)# eventlog disable
```

Syntax: eventlog disable

NOTE

This command is not saved across reloads by default. To have it take effect after a reload use the **write mem** command to save the **eventlog disable** command to the startup configuration file.

Once you have disabled event logging you can re-enable it as shown in the following.

```
ServerIronADX(config)# eventlog enable
```

Syntax: eventlog enable

NOTE

The **event log enable** command is only used where you want to enable the event logging feature that was previously disabled.

Changing the default maximum size of the event log

The default maximum size of the event log file is 32 MB. This value can be changed as shown in the following.

```
ServerIronADX(config)# eventlog size 64
```

Syntax: eventlog size <eventlog-file-size>

The <eventlog-file-size> is the size in MB that you want to change the default maximum size of the event log file to. It can be set to any value between 32 and 256.

Displaying event log status

You can display the current status of the event log as shown in the following.

```
ServerIronADX# show eventlog
Event Log is enabled
Event Log MAX specified size is 256 MB
Event Log Size is - .2fMB
```

Syntax: show eventlog

The contents of the display are defined in the following table.

TABLE 5 Display from **show eventlog** command

This field...	Displays...
Enabled	Displays if eventlog is currently enabled.
Max Size of log	The maximum size of the event log as currently configured. By default this value is 32MB. Once the maximum log size is reached, the log file is overwritten.
Current Size of log	The current size of the event log file on the internal USB drive.

Clearing the event log

You can clear the event log as shown in the following.

```
ServerIronADX# clear eventlog
```

Syntax: clear eventlog

Additional system management functions

This section contains information on system management functions.

Configuring uplink utilization lists

You can configure uplink utilization lists that display the percentage of a given uplink port's bandwidth that is used by a specific list of downlink ports. Both transmit and receive traffic are counted in each percentage.

NOTE

This feature is intended for ISP or collocation environments in which downlink ports are dedicated to various customers' traffic and are isolated from one another. If traffic regularly passes between the downlink ports, the information displayed by the utilization lists does not provide a clear depiction of traffic exchanged by the downlink ports and the uplink port.

Each uplink utilization list consists of the following:

- Utilization list number (1, 2, 3, or 4)
- One or more uplink ports
- One or more downlink ports

1 Additional system management functions

Each list displays the uplink port and the percentage of that port's bandwidth that was utilized by the downlink ports over the most recent 30-second interval. You can configure up to four bandwidth utilization lists.

To configure a link utilization list with port 1 as the uplink port and ports 2 and 3 as the downlink ports, enter a command such as the following.

```
ServerIronADX(config)# relative-utilization 1 uplink eth 1 downlink eth 2 to 3
```

Syntax: **[no] relative-utilization** *<num>* **uplink ethernet** *<portnum>* **[to** *<portnum>* **|** *<portnum>...*]
downlink ethernet *<portnum>* **[to** *<portnum>* **|** *<portnum>...*]

The *<num>* parameter specifies the list number. You can configure up to four lists. Specify a number from 1 – 4.

The **uplink ethernet** parameters and the port number(s) you specify after the parameters indicate the uplink port(s).

The **downlink ethernet** parameters and the port number(s) you specify after the parameters indicate the downlink port(s).

Displaying an uplink utilization list

Displaying an uplink utilization list allows you to observe the percentage of the uplink's bandwidth that each of the downlink ports used during the most recent 30-second port statistics interval. The number of packets sent and received between the two ports is listed, as well as the ratio of each individual downlink port's packets relative to the total number of packets on the uplink.

To display an uplink utilization list, enter a command such as the following.

```
ServerIronADX(config)# show relative-utilization 1
uplink: ethe 1
30-sec total uplink packet count = 3011
packet count ratio (%)
  1/ 2:60   1/ 3:40
```

In this example, ports 2 and 3 are sending traffic to port 1. Port 2 and port 3 are isolated (not shared by multiple clients) and typically do not exchange traffic with other ports except for the uplink port, port 1.

Syntax: **show relative-utilization** *<num>*

The *<num>* parameter specifies the list number.

Setting system time and date

You can set the system time and date for a ServerIron ADX.

Clock settings are saved over power cycles. You can also configure the system to reference a SNTP server at power up. This server will then automatically download the correct time reference for the network. For more details on this capability, reference the **sntp** command at the privileged EXEC level and the **sntp poll-interval** and **sntp server** commands at the global CONFIG level.

To set the system time and date for a ServerIron ADX, enter a command such as the following.

```
ServerIronADX# clock set 10:15:05 10-15-98
```

Syntax: **[no] clock set** *<hh:mm:ss>* *<mm-dd-yy>* **|** *<mm-dd-yyyy>*

To set the time zone, use the **clock timezone** command at the global configuration level.

Activating or deactivating daylight savings time

To automatically activate and deactivate daylight savings time for the relevant time zones, enter the following command.

```
ServerIronADX(config)# clock summer-time
```

Syntax: [no] clock summer-time

Setting the time zone

To define the time zone of the clock, enter a command such as the following.

```
ServerIronADX(config)# clock timezone us eastern
```

Syntax: [no] clock timezone **gmt** | **us** <time-zone>

The <time-zone> parameter can be any of the following:

- US time zones—alaska, aleutian, arizona, central, east-indiana, eastern, hawaii, michigan, mountain, pacific, samoa
- GMT time zones—gmt+12, gmt+11, gmt+10...gmt+01, gmt+00, gmt-01...gmt-10, gmt-11, gmt-12

The default is gmt + 00.

The **clock timezone** command is used in conjunction with the **clock set** command or for timestamps obtained from a SNTP server. The **clock set** command is configured at the privileged EXEC level of the CLI.

Use this **clock** command before all others to ensure accuracy of the clock settings. For those time zones that recognize daylight savings time, the clock summer-time command will also need to be defined.

Clock settings are not saved over power cycles; however, you can configure the system to reference a SNTP server at power up. This server will then automatically download the correct time reference for the network. The local ServerIron ADX will then adjust the time according to its time zone setting. For more details on setting up a SNTP reference clock, refer to the **sntp** command at the privileged EXEC level and the **sntp poll-interval** and **sntp server** commands at the global CONFIG level.

DST change notice for networks using US time zones

The new Daylight Saving Time (DST) change that went into effect on March 11, 2007 affects networks in the US time zones. Because of this change, your network clock might not be correct.

If your network uses US time zones, and it needs to maintain the correct time, you must enable the following command.

```
ServerIronADX(config)# clock timezone us pacific
```

Syntax: clock timezone us {**pacific** | **eastern** | **central** | **mountain**}

NOTE

This command must be configured on every device that uses the US DST.

To verify the change, use the following command.

Syntax: show clock

```
showing with RTC...  
08:23:42, GMT+00, Mon Sep 12 2011  
For more information, refer to the marketing advisory.
```

Changing the shutdown temperature

You can change the shutdown temperature of a module containing a temperature sensor. If the temperature matches or exceeds the shutdown temperature, the software sends a Syslog message to the Syslog buffer and also to the Syslog server if configured. The software also sends an SNMP trap to the SNMP trap receiver, if you have configured the device to use one.

If the temperature equals or exceeds the shutdown temperature for five consecutive polls of the temperature by the software, the software shuts down the module to prevent damage.

To change the temperature for a management module, enter the following command.

```
ServerIronADX# temperature shutdown mgmt 65
```

Syntax: temperature shutdown mgmt <value>

The <value> parameter can be 0 – 125 degrees celsius. The default is 100.

To change the temperature for a line card, enter the following command.

```
ServerIronADX# temperature shutdown lc 65
```

Syntax: temperature shutdown lc <value>

The <value> parameter can be 0 – 75 degrees celsius. The default is 75.

NOTE

The **lc** option is required only for chassis type model such as ADX4000, ADX10000, and so on.

NOTE

if you change the temperature, you must execute “write memory” to save the changed temperature.

Changing the temperature warning

You can change the warning temperature of a module containing a temperature sensor. If the temperature of the module reaches the warning value, the software sends a Syslog message to the Syslog buffer and also to the Syslog server, if configured. In addition, the software sends an SNMP trap to the SNMP trap receiver, if you have configured the device to use one.

NOTE

You cannot set the warning temperature to a value higher than the shutdown temperature.

To change the warning temperature for a management module, enter the following command.

```
ServerIronADX# temperature warning mgmt 55
```

Syntax: temperature warning mgmt <value>

The <value> parameter can be 0 – 125. The default is 85.

To change the warning temperature for a line card, enter the following command.

```
ServerIronADX# temperature warning lc 55
```

Syntax: `temperature warning lc <value>`

The `<value>` parameter can be 0 – 65. The default is 65.

NOTE

The `lc` option is required only for chassis type model such as ADX4000, ADX10000, and so on.

NOTE

if you change the temperature, you must execute “write memory” to save the changed temperature.

Changing the number of seconds between polls

You can change the number of seconds between polls of the power supply and fan status, by entering a command such as the following.

```
ServerIronADX(config)# chassis poll-time 30
```

Syntax: `[no] chassis poll-time <num>`

The default is 60 seconds.

Use the **show chassis** command to display the hardware status.

Disabling or re-enabling status polling

You can disable or re-enable status polling for individual power supplies and fans. When you disable status polling, a fault in the power supply does not generate a trap in the system log.

To disable status polling, enter the following command.

```
ServerIronADX(config)# no chassis trap-log ps2
```

Syntax: `[no] chassis trap-log ps1 | ps2 | fan1 | fan2`

Adjusting inter-packet gap

You can adjust the inter-packet gap (IPG) to match older adapters that do not meet the default IPG requirements for Ethernet.

NOTE

Entering the value of 0 within the `ipg10`, `ipg100`, and `ipg1000` commands restore the IPG to the default of 12 bytes.

Modifying the IPG on 10Mbps Ethernet segment

You can modify the inter-packet gap (delay) between packets on a 10Mbps Ethernet segment.

In determining the value to enter in the CLI command, note that one byte equals .8 microseconds for packets on a 10Mbps segment, so the following equation can be used.

$$\text{IPG10} = 9.6 \text{ microseconds} + (\text{value} * .8), \text{ where value is the number of bytes by which you want to increase the inter-packet gap.}$$

To increase the delay between packets by 3.2 microseconds, enter commands such as the following.

```
ServerIronADX(config) #int e 4
ServerIronADX(config-if-4) # ipg10 4
```

Syntax: [no] **ipg10** <value>

The <value> parameter is 0 – 100 bytes. By default, the delay between packets will be 12 bytes (9.6 microseconds, **ipg10 0**).

Modifying the IPG on 100Mbps Ethernet segment

You can modify the inter-packet gap (delay) between packets on a 100Mbps Ethernet segment on a port-by-port basis. You do this only to adjust the inter-packet gap to match that of older adapters that do not meet the default IPG requirements for Fast Ethernet.

In determining the value to enter in the CLI command, note that one byte equals .08 microseconds for packets on a 100Mbps segment, so the following equation can be used.

$IPG100 = 0.96 \text{ microseconds} + (\text{value} * .08)$, where value is the number of bytes by which you want to increase the inter-packet gap.

To increase the delay between packets by 3.2 microseconds, enter the port to be modified and then enter the value of 40 ($40 * .08 = 3.2$ microseconds), such as the following.

```
ServerIronADX(config) # int e 3
ServerIronADX(config-if-3) # ipg100 40
```

Syntax: **ipg100** <value>

The <value> parameter is 0 to 100. By default, the delay between packets will be 12 bytes (**ipg100 0**, 0.96 microseconds).

Modifying the IPG on 1000Mbps Gigabit Ethernet segment

You can modify the inter-packet gap (delay) between packets on a 1000Mbps Gigabit Ethernet segment on a port-by-port basis. You do this only to adjust the inter-packet gap to match that of older adapters that do not meet the default IPG requirements for Gigabit Ethernet.

In determining the value to enter in the CLI command, note that one byte equals .008 microseconds for packets on a 1000Mbps segment, so the following equation can be used.

$IPG1000 = .096 \text{ microseconds} + (\text{value} * .008)$, where value is the number of bytes by which you want to increase the inter-packet gap.

To increase the delay between packets by .32 microseconds, first enter the port to be modified and then enter the value of 40 ($40 * .008 = .32$ microseconds), such as the following.

```
ServerIronADX(config) # int e 3
ServerIronADX(config-if-3) # ipg1000 40
```

Syntax: **ipg1000** <value>

By default, the delay between packets will be 12 bytes (.096 microseconds, **ipg1000 0**).

Assigning a port name

To assign a name to an interface, which provides additional identification for a segment on the network, enter commands such as the following.

```
ServerIronADX(config)# interface e 1
ServerIronADX(config-if-1)# port-name marketing-funk
```

Syntax: [no] port-name <text>

Modifying port speed and duplex mode

You can modify the port speed and duplex mode for 10BaseT and 100BaseTx ports.

Gigabit (1000BaseSx and 1000BaseLx) and 100BaseFx ports operate at a fixed speed and mode (full-duplex) and cannot be modified.

To modify the port speed and duplex mode for a 10BaseT or 100BaseTx port, enter commands such as the following.

```
ServerIronADX(config)# interface e8
ServerIronADX(config-if-8)# speed-duplex 10-full
```

Syntax: [no] speed-duplex <value>

The <value> parameter can be 10-full, 10-half, 100-full, 100-half, or auto. The default is 10/100 auto.

Enabling a mirror port

You can enable and assign a specific port to operate as a mirror port for other ports on a ServerIron ADX, by using the **mirror-port ethernet** command. Once enabled, you can connect an external traffic analyzer to the port for traffic analysis.

You also need to enable the **monitor** command on a port for it to be mirrored by this port.

To assign port 1 as the mirror port and port 5 as the port to be monitored, enter commands such as the following.

```
ServerIronADX(config)# mirror-port e 1/1
ServerIronADX(config)# interface e 5/2
ServerIronADX(config-if)# monitor both
```

Syntax: [no] mirror-port ethernet <portnum>

Syntax: [no] monitor input | output | both

Use **monitor input | output | both** to select a port to be diagnosed by a designated mirror port. You can configure incoming, outgoing or both incoming and outgoing traffic to be monitored on the port.

```
ServerIronADX(config)# int e 2/1
ServerIronADX(config-if-e100-2/1)# monitor ethernet 2/1 ?
  both      Both incoming and outgoing packets
  input     Incoming packets
  output    Outgoing packets
```

Displaying port mirroring and monitoring information

The mirror port feature lets you connect a protocol analyzer to a port on a Brocade device to observe the traffic flowing into and out of another port on the same device. To use this feature, you specify the port you want to monitor and the port into which you are plugging the protocol analyzer.

NOTE

ServerIron supports more than one active mirror port at a time. By default, no mirror port is assigned.

To display the current port mirroring and monitoring configuration, enter the following command.

```
ServerIronADX(config)# show monitor
Mirror Interface:    ethernet 4/1
Monitored Interfaces:
  Both    Input    Output
-----
ethernet 4/3
```

In this example, port 4/1 is the mirror interface, to which the software copies (“mirrors”) the traffic on port 4/3. In this case, both directions of traffic on the monitored port are mirrored to port 4/1.

Syntax: show monitor

If only the incoming traffic is mirrored, the monitored interface is listed under Input. If only the outbound traffic is mirrored, the monitored interface is listed under Output.

Setting QoS priority

You can set the Quality-of-Service (QoS) priority level for a port, VLAN, static MAC address, or Layer 4 session. You can select the **normal** queue or the **high** priority queue. All traffic is in the **normal** queue by default. When you allocate a port, VLAN, static MAC address, or Layer 4 session to the high-priority queue, all traffic queued up for that item is processed before any traffic in the normal queue for the same item is processed.

QoS applies to outbound traffic only.

To set the QoS priority level, enter commands such as the following.

```
ServerIronADX(config)# interface e 6
ServerIronADX(config-if-6)# qos-priority high
```

Syntax: [no] qos-priority normal | high

Turning the flow control on or off

By default, flow control is on. To turn flow control (802.3x) for full-duplex ports on or off, enter the following command.

```
ServerIronADX(config)# no flow-control
```

Or assign it to a specific interface.

```
ServerIronADX(config)# int e5/2
ServerIronADX(config-if-5/2)#no flow control
```

Syntax: [no] flow-control

Setting the negotiation mode

You can change the default negotiation mode for Gigabit ports on Chassis devices, by using the **gig-default** command. It enables 802.3z negotiation for gigabit over optical fiber. Both sides of the circuit need to be configured with this feature.

If you enter **auto-gig**, then **gig-default auto-gig** is added to the running config.

NOTE

802.3x is flow-control over full-duplex regardless of speed. Half duplex flow control uses backpressure. 802.3z is gigabit fiber. 802.3ab is gigabit copper.

To set a negotiation mode to off, enter the following command.

```
ServerIronADX(config)# gig-default neg-off
```

To override the global setting and set the negotiation mode to auto-Gigabit for ports 4/1 – 4/4, enter commands such as the following.

```
ServerIronADX(config)# int ethernet 4/1 to 4/4
ServerIronADX(config-mif-4/1-4/4)# gig-default auto-gig
```

Syntax: [no] **gig-default neg-full-auto | auto-gig | neg-off**

The **neg-full-auto** option specifies that the port first tries to perform a handshake with the other port to exchange capability information. If the other port does not respond to the handshake attempt, the port uses the manually configured configuration information (or the defaults if an administrator has not set the information). This is the default for Chassis devices.

The **auto-gig** option specifies that the port tries to perform a handshake with the other port to exchange capability information. This is still the default for Stackable devices.

The **neg-off** option specifies that the port does not try to perform a handshake. Instead, the port uses configuration information manually configured by an administrator.

Defining the performance mode

You can define the performance mode as 'high' to allow flow control to activate at an earlier stage, when heavy congestion exists on the network. This feature must be saved to memory and the system reset before it becomes active.

To define the performance mode as high, enter commands such as the following.

```
ServerIronADX(config)# perf-mode hi
Reload required. Please write memory and then reload or power cycle.
ServerIronADX(config)# write memory
.Write startup-config in progress.
.Write startup-config done.
ServerIronADX(config) # exit
ServerIronADX# reload
```

Syntax: [no] **perf-mode**

Forwarding Layer 2 and Layer 3 pass-through traffic to the CPU

By default, the ServerIron ADX forwards Layer 2 and Layer 3 pass-through traffic in hardware. It forwards Layer 4 – Layer 7 pass-through traffic to the CPU for processing.

1 Remapping processing for a forwarding module to a BP

You can configure the device to forward Layer 2 and Layer 3 pass-through traffic to the CPU for processing, instead of processing it in hardware. To do this, enter the following command.

```
ServerIronADX(config)# server cpu-forward
```

Syntax: [no] server cpu-forward

Hardware forwarding for non L4-7 traffic flows

The ServerIron ADX supports hardware forwarding of pass-through traffic in SLB and TCS modes. The ServerIron ADX can be configured so that only Layer 4-7 traffic is processed by the CPU, and pass-through traffic is forwarded in hardware.

Hardware forwarding is enabled by default in the following configurations:

- SLB only
- VIP SYN defense or VIP SYN proxy
- TCS only
- TCS cache-enabled
- Any combination of the above features

If any ServerIron ADX features other than those listed above are enabled on the device, then hardware forwarding is disabled by default.

NOTES:

- If TCS is configured without source NAT and destination NAT, hardware forwarding is disabled for the DMA to which the cache server is connected, if the ServerIron ADX can determine this DMA. If the ServerIron ADX cannot determine this DMA, hardware forwarding is disabled for all DMAs.
- Hardware forwarding is disabled if the TCS port is FTP, MMS, or RTSP.
- Hardware forwarding is disabled if the TCS Cache Server Spoofing feature is enabled.
- Pass-through traffic with a destination address that has the same last 13 bits as a real server, cache server, or VIP may be treated as Layer 4-7 traffic and sent to the CPU for processing.
- If CAM space is exhausted on the device, then hardware forwarding is disabled, and all packets are sent to the CPU. While this situation is rare, it can occur in configurations that have a large number of real servers or VIPs.
- Fragmented TCS packets may, in rare instances, be hardware forwarded.
- SLB DSR reverse traffic hardware forwarding is disabled if TCS cache-enable is ON.
- VIP-host-range is not supported.

Remapping processing for a forwarding module to a BP

You can remap processing for a forwarding module to a specific BP.

Brocade recommends that you change slot allocations only if Brocade technical support advises the change or the documentation for a feature states that the change is required.

To remap processing for a forwarding module to a specific BP, enter a command such as the following.


```
ServerIronADX(config)# wsm wsm-map slot 3 wsm-slot 2 wsm-cpu 1
```

This command remaps processing for the forwarding module in slot 3 to BP 1 on the Application Switching Module in slot 2.

Syntax: **[no] wsm wsm-map** *<from-slotnum>* **wsm-slot** *<to-slotnum>* **wsm-cpu** *<cpunum>*

The *<from-slotnum>* parameter specifies the slot that contains the forwarding module.

The *<to-slotnum>* parameter specifies the slot that contains the Application Switching Module.

The *<cpunum>* parameter specifies the BP on *<to-slotnum>* that will perform the processing. The BPs are numbered from 1 – 3.

Specifying the maximum number of unknown unicast packets

You can specify the maximum number of unknown-unicast packets the device can forward each second. By default the device sends unknown unicasts and all other traffic at wire speed and is limited only by the capacities of the hardware. However, if other devices in the network cannot handle unlimited unknown-unicast traffic, this command allows you to relieve those devices by throttling the unknown unicasts at the Brocade device.

The unknown-unicast limit does not affect broadcast or multicast traffic. However, you can use the **broadcast limit** and **multicast limit** commands to control these types of traffic.

To specify the maximum number of unknown-unicast packets the device can forward each second, enter commands such as the following.

```
ServerIronADX(config)# interface e8
ServerIronADX(config-if-8)# unknown-unicast limit 30000
```

Syntax: **[no] unknown-unicast limit** *<num>*

1 Specifying the maximum number of unknown unicast packets

Secure Access Management

Securing access methods

The following table lists the management access methods available on a ServerIron, how they are secured by default, and the ways in which they can be secured.

TABLE 6 Ways to secure management access to ServerIrons

Access method	How the access method is secured by default	Ways to secure the access method	See page
Serial access to the CLI	Not secured	Establish passwords for management privilege levels	page 100
Access to the Privileged EXEC and CONFIG levels of the CLI	Not secured	Establish a password for Telnet access to the CLI	page 100
		Establish passwords for management privilege levels	page 100
		Set up local user accounts	page 104
		Configure TACACS or TACACS+ security	page 106
		Configure RADIUS security	page 122
Telnet access IPv4 and IPv6 addresses	Not secured	Regulate Telnet access using ACLs	page 93
		Allow Telnet access only from specific IP addresses	page 95
		Allow Telnet access only to clients connected to a specific VLAN	page 96
		Disable Telnet access	page 98
		Establish a password for Telnet access	page 100
		Establish passwords for privilege levels of the CLI	page 100
		Set up local user accounts	page 104
		Configure TACACS or TACACS+ security	page 106
		Configure RADIUS security	page 122
Secure Shell (SSH) access IPv4 and IPv6 addresses for SSHv2	Not configured	Configure SSH	
		Regulate SSH access using ACLs	page 93
		Establish passwords for privilege levels of the CLI	page 100
		Set up local user accounts	page 104
		Configure TACACS or TACACS+ security	page 106
		Configure RADIUS security	page 122

TABLE 6 Ways to secure management access to ServerIrons (Continued)

Access method	How the access method is secured by default	Ways to secure the access method	See page
Web management access IPv4 and IPv6 addresses	SNMP read or read-write community strings	Regulate Web management access using ACLs	page 94
		Allow Web management access only from specific IP addresses	page 95
		Allow Web management access only to clients connected to a specific VLAN	page 97
		Disable Web management access	page 99
		Set up local user accounts	page 104
		Establish SNMP read or read-write community strings for SNMP versions 1 and 2	page 137
		Establishing user groups for SNMP version 3	
		Configure TACACS or TACACS+ security	page 106
SNMP (IronView) access IPv4 addresses	SNMP read or read-write community strings and the password to the Super User privilege level Note: SNMP read or read-write community strings are always required for SNMP access to the device.	Configure RADIUS security	page 122
		Regulate SNMP access using ACLs	page 94
		Allow SNMP access only from specific IP addresses	page 96
		Disable SNMP access	page 92
		Allow SNMP access only to clients connected to a specific VLAN	page 97
		Establish passwords to management levels of the CLI	page 100
		Set up local user accounts	page 104
TFTP access IPv4 and IPv6 addresses	Not secured	Establish SNMP read or read-write community strings	page 106
		Allow TFTP access only to clients connected to a specific VLAN	page 97

Restricting remote access to management functions

You can restrict access to management functions from remote sources, including Telnet, the Web Management Interface, and SNMP. The following methods for restricting remote access are supported:

- Using ACLs to restrict Telnet, Web Management Interface, or SNMP access
- Allowing remote access only from specific IP addresses
- Allowing remote access only to clients connected to a specific VLAN
- Specifically disabling Telnet or Web Management Interface access to the device

The following sections describe how to restrict remote access to a ServerIron using these methods.

Using ACLs to restrict remote access

You can use standard ACLs to control the following access methods to management functions on a ServerIron:

- Telnet access
- SSH access
- Web management access
- SNMP access

To configure access control for these management access methods.

1. Configure an ACL with the IP addresses you want to allow to access the device
2. Configure a Telnet access group, SSH access group, web access group, and SNMP community strings. Each of these configuration items accepts an ACL as a parameter. The ACL contains entries that identify the IP addresses that can use the access method.

The following sections present examples of how to secure management access using ACLs.

Using an ACL to restrict Telnet access

To configure an ACL that restricts Telnet access to the device, enter commands such as the following.

```
ServerIronADX(config)# access-list 10 deny host 209.157.22.32 log
ServerIronADX(config)# access-list 10 deny 209.157.23.0 0.0.0.255 log
ServerIronADX(config)# access-list 10 deny 209.157.24.0 0.0.0.255 log
ServerIronADX(config)# access-list 10 deny 209.157.25.0/24 log
ServerIronADX(config)# access-list 10 permit any
ServerIronADX(config)# telnet access-group 10
ServerIronADX(config)# write memory
```

Syntax: `telnet access-group <num>`

The `<num>` parameter specifies the number of a standard ACL and must be from 1 – 99.

The commands above configure ACL 10, then apply the ACL as the access list for Telnet access. The device allows Telnet access to all IP addresses except those listed in ACL 10.

To configure a more restrictive ACL, create permit entries and omit the **permit any** entry at the end of the ACL.

Example

```
ServerIronADX(config)# access-list 10 permit host 209.157.22.32
ServerIronADX(config)# access-list 10 permit 209.157.23.0 0.0.0.255
ServerIronADX(config)# access-list 10 permit 209.157.24.0 0.0.0.255
ServerIronADX(config)# access-list 10 permit 209.157.25.0/24
ServerIronADX(config)# telnet access-group 10
ServerIronADX(config)# write memory
```

The ACL in this example permits Telnet access only to the IP addresses in the **permit** entries and denies Telnet access from all other IP addresses.

Using an ACL to restrict SSH access

To configure an ACL that restricts SSH access to the device, enter commands such as the following.

2 Restricting remote access to management functions

```
ServerIronADX(config)# access-list 12 deny host 209.157.22.98 log
ServerIronADX(config)# access-list 12 deny 209.157.23.0 0.0.0.255 log
ServerIronADX(config)# access-list 12 deny 209.157.24.0/24 log
ServerIronADX(config)# access-list 12 permit any
ServerIronADX(config)# ssh access-group 12
ServerIronADX(config)# write memory
```

Syntax: `ssh access-group <num>`

The `<num>` parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACL 12, then apply the ACL as the access list for SSH access. The device denies SSH access from the IP addresses listed in ACL 12 and permits SSH access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny SSH access from all IP addresses.

NOTE

In this example, the command `ssh access-group 10` could have been used to apply the ACL configured in the example for Telnet access. You can use the same ACL multiple times.

Using an ACL to restrict Web management access

To configure an ACL that restricts Web management access to the device, enter commands such as the following.

```
ServerIronADX(config)# access-list 12 deny host 209.157.22.98 log
ServerIronADX(config)# access-list 12 deny 209.157.23.0 0.0.0.255 log
ServerIronADX(config)# access-list 12 deny 209.157.24.0/24 log
ServerIronADX(config)# access-list 12 permit any
ServerIronADX(config)# web access-group 12
ServerIronADX(config)# write memory
```

Syntax: `web access-group <num>`

The `<num>` parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACL 12, then apply the ACL as the access list for Web management access. The device denies Web management access from the IP addresses listed in ACL 12 and permits Web management access from all other IP addresses. Without the last ACL entry for permitting all packets, this ACL would deny Web management access from all IP addresses.

Using ACLs to restrict SNMP access

To restrict SNMP access to the device using ACLs, enter commands such as the following.

NOTE

The syntax for using ACLs for SNMP access is different from the syntax for controlling Telnet, SSH, and Web management access using ACLs.

```
ServerIronADX(config)# access-list 25 deny host 209.157.22.98 log
ServerIronADX(config)# access-list 25 deny 209.157.23.0 0.0.0.255 log
ServerIronADX(config)# access-list 25 deny 209.157.24.0 0.0.0.255 log
ServerIronADX(config)# access-list 30 deny 209.157.25.0 0.0.0.255 log
ServerIronADX(config)# access-list 30 deny 209.157.26.0/24 log
ServerIronADX(config)# access-list 30 permit any
ServerIronADX(config)# snmp-server community public ro 25
ServerIronADX(config)# snmp-server community private rw 30
ServerIronADX(config)# write memory
```

Syntax: `snmp-server community <string> ro | rw <num>`

The `<string>` parameter specifies the SNMP community string the user must enter to gain SNMP access.

The `ro` parameter indicates that the community string is for read-only ("get") access. The `rw` parameter indicates the community string is for read-write ("set") access.

The `<num>` parameter specifies the number of a standard ACL and must be from 1 – 99.

These commands configure ACLs 25 and 30, then apply the ACLs to community strings.

ACL 25 is used to control read-only access using the "public" community string. ACL 30 is used to control read-write access using the "private" community string.

Restricting remote access to the device to specific IP addresses

By default, a ServerIron does not control remote management access based on the IP address of the managing device. You can restrict remote management access to a single IP address for the following access methods:

- Telnet access
- Web management access
- SNMP access

In addition, if you want to restrict all three access methods to the same IP address, you can do so using a single command.

The following examples show the CLI commands for restricting remote access. You can specify only one IP address with each command. However, you can enter each command ten times to specify up to ten IP addresses.

NOTE

You cannot restrict remote management access using the Web Management Interface.

Restricting Telnet access to a specific IP address

To allow Telnet access to the ServerIron only to the host with IP address 209.157.22.39, enter the following command.

```
ServerIronADX(config)# telnet-client 209.157.22.39
```

Syntax: `[no] telnet-client <ip-addr>`

Restricting Web management access to a specific IP address

To allow Web management access to the ServerIron only to the host with IP address 209.157.22.26, enter the following command.

```
ServerIronADX(config)# web-client 209.157.22.26
```

Syntax: `[no] web-client <ip-addr>`

Restricting SNMP access to a specific IP address

To allow SNMP access (which includes IronView) to the ServerIron only to the host with IP address 209.157.22.14, enter the following command.

```
ServerIronADX(config)# snmp-client 209.157.22.14
```

Syntax: [no] snmp-client <ip-addr>

Restricting all remote management access to a specific IP address

To allow Telnet, Web, and SNMP management access to the ServerIron only to the host with IP address 209.157.22.69, you can enter three separate commands (one for each access type) or you can enter the following command.

```
ServerIronADX(config)# all-client 209.157.22.69
```

Syntax: [no] all-client <ip-addr>

Restricting remote access to the device to specific VLAN IDs

You can restrict management access to a ServerIron to ports within a specific port-based VLAN. VLAN-based access control applies to the following access methods:

- Telnet access
- Web management access
- SNMP access
- TFTP access

By default, access is allowed for all the methods listed above on all ports. Once you configure security for a given access method based on VLAN ID, access to the device using that method is restricted to only the ports within the specified VLAN.

VLAN-based access control works in conjunction with other access control methods. For example, suppose you configure an ACL to permit Telnet access only to specific client IP addresses, and you also configure VLAN-based access control for Telnet access. In this case, the only Telnet clients that can access the device are clients that have one of the IP addresses permitted by the ACL *and* are connected to a port that is in a permitted VLAN. Clients who have a permitted IP address but are connected to a port in a VLAN that is not permitted still cannot access the device through Telnet.

Restricting Telnet access to a specific VLAN

To allow Telnet access only to clients in a specific VLAN, enter a command such as the following.

```
ServerIronADX(config)# telnet server enable vlan 10
```

The command in this example configures the device to allow Telnet management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

Syntax: [no] telnet server enable vlan <vlan-id>

Restricting Web management access to a specific VLAN

To allow Web management access only to clients in a specific VLAN, enter a command such as the following.

```
ServerIronADX(config)# web-management enable vlan 10
```

The command in this example configures the device to allow Web management access only to clients connected to ports within port-based VLAN 10. Clients connected to ports that are not in VLAN 10 are denied management access.

Syntax: [no] web-management enable vlan <vlan-id>

Restricting SNMP access to a specific VLAN

To allow SNMP access only to clients in a specific VLAN, enter a command such as the following.

```
ServerIronADX(config)# snmp-server enable vlan 40
```

The command in this example configures the device to allow SNMP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] snmp-server enable vlan <vlan-id>

Restricting TFTP access to a specific VLAN

To allow TFTP access only to clients in a specific VLAN, enter a command such as the following.

```
ServerIronADX(config)# tftp client enable vlan 40
```

The command in this example configures the device to allow TFTP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

Syntax: [no] tftp client enable vlan <vlan-id>

Designated VLAN for Telnet management sessions to a Layer 2 Switch

By default, the management IP address you configure on a Layer 2 Switch applies globally to all the ports on the device. This is true even if you divide the device's ports into multiple port-based VLANs.

By default, the management IP address you configure on a Layer 2 Switch applies globally to all the ports on the device. This is true even if you divide the device's ports into multiple port-based VLANs.

If you want to restrict the IP management address to a specific port-based VLAN, you can make that VLAN the designated management VLAN for the device. When you configure a VLAN to be the designated management VLAN, the management IP address you configure on the device is associated only with the ports in the designated VLAN. To establish a Telnet management session with the device, a user must access the device through one of the ports in the designated VLAN.

You also can configure up to five default gateways for the designated VLAN, and associate a metric with each one. The software uses the gateway with the lowest metric. The other gateways reside in the configuration but are not used. You can use one of the other gateways by modifying the configuration so that the gateway you want to use has the lowest metric.

2 Restricting remote access to management functions

If more than one gateway has the lowest metric, the software uses the gateway that appears first in the running-config.

NOTE

If you have already configured a default gateway globally and you do not configure a gateway in the VLAN, the software uses the globally configured gateway and gives the gateway a metric value of 1.

To configure a designated management VLAN, enter commands such as the following.

```
ServerIronADX(config)# vlan 10 by port
ServerIronADX(config-vlan-10)# untag ethernet 1/1 to 1/4
ServerIronADX(config-vlan-10)# management-vlan
ServerIronADX(config-vlan-10)# default-gateway 10.10.10.1 1
ServerIronADX(config-vlan-10)# default-gateway 20.20.20.1 2
```

These commands configure port-based VLAN 10 to consist of ports 1/1 – 1/4 and to be the designated management VLAN. The management VLAN command is not supported on Layer 2 code. The last two commands configure default gateways for the VLAN. Since the 10.10.10.1 gateway has a lower metric, the software uses this gateway. The other gateway remains in the configuration but is not used. You can use the other one by changing the metrics so that the 20.20.20.1 gateway has the lower metric.

Syntax: [no] management-vlan

Syntax: [no] default-gateway <ip-addr> <metric>

The <ip-addr> parameters specify the IP address of the gateway router.

The <metric> parameter specifies the metric (cost) of the gateway. You can specify a value from 1 – 5. There is no default. The software uses the gateway with the lowest metric.

Disabling specific access methods

You can specifically disable the following access methods:

- Telnet access
- Web management access

NOTE

If you disable Telnet access, you will not be able to access the CLI except through a serial connection to the management module. If you disable SNMP access, you will not be able to use [IronView](#) or [third-party](#) SNMP management applications.

Disabling Telnet access

Telnet access is enabled by default. You can use a Telnet client to access the CLI on the device over the network. If you do not plan to use the CLI over the network and want to disable Telnet access to prevent others from establishing CLI sessions with the device, enter the following command.

```
ServerIronADX(config)# no telnet-server
```

To re-enable Telnet operation, enter the following command.

```
ServerIronADX(config)# telnet-server
```

Syntax: [no] telnet-server

Disabling Web management access

If you want to prevent access to the device through the Web Management Interface, you can disable the Web Management Interface.

NOTE

As soon as you make this change, the device stops responding to Web management sessions. If you make this change using your Web browser, your browser can contact the device, but the device will not reply once the change takes place.

To disable the Web Management Interface, enter the following command.

```
ServerIronADX(config)# no web-management
```

To re-enable the Web Management Interface, enter the following command.

```
ServerIronADX(config)# web-management
```

Syntax: [no] web-management

Disabling Web management access by HP TOP-TOOLS

By default, TCP ports 80 and 280 are enabled on the ServerIron. TCP port 80 (HTTP) allows access to the device's Web Management Interface. TCP port 280 allows access to the device by HP TOP-TOOLS.

The **no web-management** command disables both TCP ports. However, if you want to disable only port 280 and leave port 80 enabled, use the **hp-top-tools** option with the command.

Example

```
ServerIronADX(config)# no web-management hp-top-tools
```

Syntax: [no] web-management [allow-no-password | enable [vlan <vlan-id>] | front-panel | hp-top-tools | list-menu]

The **hp-top-tools** parameter disables TCP port 280.

Setting passwords

Passwords can be used to secure the following access methods:

- Telnet access can be secured by setting a Telnet password. Refer to [“Setting a Telnet password”](#) on page 100.
- Access to the Privileged EXEC and CONFIG levels of the CLI can be secured by setting passwords for management privilege levels. Refer to [“Setting passwords for management privilege levels”](#) on page 100.

This section also provides procedures for enhancing management privilege levels, recovering from a lost password, and disabling password encryption.

NOTE

You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account a management privilege level. Refer to [“Setting up local user accounts”](#) on page 104.

Setting a Telnet password

By default, the device does not require a user name or password when you log in to the CLI using Telnet.

To set the password "letmein" for Telnet access to the CLI, enter the following command at the global CONFIG level.

```
ServerIronADX(config)# enable telnet password letmein
```

Syntax: [no] enable telnet password <string>

NOTE

Any user who knows telnet password can login. User validation does not occur here.

Suppressing Telnet connection rejection messages

By default, if a ServerIron denies Telnet management access to the device, the software sends a message to the denied Telnet client. You can optionally suppress the rejection message. When you enable the option, a denied Telnet client does not receive a message from the ServerIron. Instead, the denied client simply does not gain access.

To suppress the connection rejection message, use the following CLI method.

To suppress the connection rejection message sent by the device to a denied Telnet client, enter the following command at the global CONFIG level of the CLI.

```
ServerIronADX(config)# telnet server suppress-reject-message
```

Syntax: [no] telnet server suppress-reject-message

Setting passwords for management privilege levels

You can set one password for each of the following management privilege levels:

- **Super User level** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords.
- **Port Configuration level** – Allows read-and-write access for specific ports but not for global (system-wide) parameters.
- **Read Only level** – Allows access to the Privileged EXEC mode and CONFIG mode of the CLI but only with read access.

You can assign a password to each management privilege level. You also can configure up to 16 user accounts consisting of a user name and password, and assign each user account to one of the three privilege levels. Refer to ["Setting up local user accounts"](#) on page 104.

NOTE

You must use the CLI to assign a password for management privilege levels. You cannot assign a password using the Web Management Interface.

If you configure user accounts in addition to privilege level passwords, the device will validate a user's access attempt using one or both methods (local user account or privilege level password), depending on the order you specify in the authentication-method lists. Refer to ["Configuring authentication-method lists"](#) on page 136.

To set passwords for management privilege levels.

1. At the opening CLI prompt, enter the following command to change to the Privileged level of the EXEC mode.

```
ServerIronADX> enable
ServerIronADX#
```

2. Access the CONFIG level of the CLI by entering the following command.

```
ServerIronADX# configure terminal
ServerIronADX(config)#
```

3. Enter the following command to set the Super User level password.

```
ServerIronADX(config)# enable super-user-password <text>
```

NOTE

You must set the Super User level password before you can set other types of passwords. The Super User level password can be an alphanumeric string, but cannot begin with a number.

4. Enter the following commands to set the Port Configuration level and Read Only level passwords.

```
ServerIronADX(config)# enable port-config-password <text>
ServerIronADX(config)# enable read-only-password <text>
```

NOTE

If you forget your Super User level password, refer to ["Recovering from a lost password"](#) on page 102.

Augmenting management privilege levels

Each management privilege level provides access to specific areas of the CLI by default:

- Super User level provides access to all commands and displays.
- Port Configuration level gives access to:
 - The User EXEC and Privileged EXEC levels
 - The port-specific parts of the CONFIG level
 - All interface configuration levels
- Read Only level gives access to:
 - The User EXEC and Privileged EXEC levels

You can grant additional access to a privilege level on an individual command basis. To grant the additional access, you specify the privilege level you are enhancing, the CLI level that contains the command, and the individual command.

NOTE

This feature applies only to management privilege levels on the CLI. You cannot augment management access levels for the Web Management Interface.

To enhance the Port Configuration privilege level so users also can enter IP commands at the global CONFIG level, enter a command such as the following.

```
ServerIronADX(config)# privilege configure level 4 ip
```

In this command, **configure** specifies that the enhanced access is for a command at the global CONFIG level of the CLI. The **level 4** parameter indicates that the enhanced access is for management privilege level 4 (Port Configuration). All users with Port Configuration privileges will have the enhanced access. The **ip** parameter indicates that the enhanced access is for the **IP** commands. Users who log in with valid Port Configuration level user names and passwords can enter commands that begin with "ip" at the global CONFIG level.

Syntax: `[no] privilege <cli-level> level <privilege-level> <command-string>`

The `<cli-level>` parameter specifies the CLI level and can be one of the following values.

exec – EXEC level; for example, `ServerIron> or ServerIron#`

configure – CONFIG level; for example, `ServerIron(config)#`

interface – Interface level; for example, `ServerIron(config-if-6)#`

virtual-interface – Virtual-interface level; for example, `ServerIron(config-vif-6)#`

rip-router – RIP router level; for example, `ServerIron(config-rip-router)#`

ospf-router – OSPF router level; for example, `ServerIron(config-ospf-router)#`

port-vlan – Port-based VLAN level; for example, `ServerIron(config-vlan)#`

protocol-vlan – Protocol-based VLAN level

The `<privilege-level>` indicates the number of the management privilege level you are augmenting. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The `<command-string>` parameter specifies the command you are allowing users with the specified privilege level to enter. To display a list of the commands at a CLI level, enter "?" at that level's command prompt.

Recovering from a lost password

NOTE

You can perform this procedure only from the console.

Recovery from a lost password requires direct access to a system console and a system reset. You need to configure the system to ignore the saved configuration and to use the system default. When the system boots up with the default configuration, use username **admin** and password **brocade** to get access to the console. Change the user password, and the super-user password if necessary, and reload the box after saving the configuration.

To recover from a lost password, follow these steps.

1. Start a CLI session using the console.
2. Reboot the device.
3. At the initial boot prompt at system startup, enter **b** to enter the boot monitor mode.
4. Enter **use default config** at the prompt.

NOTE

You cannot abbreviate this command. This command causes the device to ignore saved config.

5. Enter **boot system flash primary** at the prompt.
6. After the login prompt appears, use user name **admin** and password **brocade** to gain access to the Exec Mode.
7. Enter **enable** to gain access to the privileged mode.
8. Copy the startup configuration into the running configuration, by copying the startup configuration to a tftp-server and copying the same file from tftp to the running configuration.
9. Change the user password, and super-user password if necessary, and save the configuration.

Displaying the SNMP community string

If you want to display the SNMP community string, enter the following commands.

```
ServerIronADX(config)# enable password-display
ServerIronADX(config)# show snmp server
Contact:
    Location:
Community(ro): .....
```

Traps

```

Cold start: Enable
Link up: Enable
Link down: Enable
Authentication: Enable
Locked address violation: Enable
Power supply state change: Enable
Fan failure: Enable
Redundant module state change: Enable
Temperature warning: Enable
STP new root: Enable
STP topology change: Enable
L4 maximum session limit: Enable
L4 TCP SYN limit: Enable
L4 server max connection: Enable
L4 server up: Enable
L4 server down: Enable
L4 server port up: Enable
L4 server port down: Enable
L4 switch standby: Enable
L4 switch active: Enable
HTTP match-list: Enable
```

The **enable password-display** command enables display of the community string, but only in the output of the **show snmp server** command. Display of the string is still encrypted in the startup-config file and running-config. Enter the command at the global CONFIG level of the CLI.

Disabling password encryption

When you configure a password, then save the configuration to the ServerIron's flash memory, the password is also saved to flash as part of the configuration file. By default, the passwords are encrypted so that the passwords cannot be observed by another user who displays the configuration file. Even if someone observes the file while it is being transmitted over TFTP, the password is encrypted.

If you want to remove the password encryption, you can disable encryption by entering the following command.

```
ServerIronADX(config)# no service password-encryption
```

Syntax: [no] service password-encryption

Specifying a minimum password length

By default, the ServerIron imposes no minimum length on the Line (Telnet), Enable, or Local passwords. You can configure the device to require that Line, Enable, and Local passwords be at least a specified length.

For example, to specify that the Line, Enable, and Local passwords be at least 8 characters, enter the following command.

```
ServerIronADX(config)# enable password-min-length 8
```

Syntax: enable password-min-length <number-of-characters>

The <number-of-characters> can be from 1 – 48.

Setting up local user accounts

You can define up to 16 local user accounts on a ServerIron. User accounts regulate who can access the management functions in the CLI using the following methods:

- Telnet access
- Web management access
- SNMP access

Local user accounts provide greater flexibility for controlling management access to ServerIrons than do management privilege level passwords and SNMP community strings of SNMP versions 1 and 2. You can continue to use the privilege level passwords and the SNMP community strings as additional means of access authentication. Alternatively, you can choose not to use local user accounts and instead continue to use only the privilege level passwords and SNMP community strings. Local user accounts are backward-compatible with configuration files that contain privilege level passwords. Refer to [“Setting passwords for management privilege levels”](#) on page 100.

If you configure local user accounts, you also need to configure an authentication-method list for Telnet access, Web management access, and SNMP access. Refer to [“Configuring authentication-method lists”](#) on page 136.

For each local user account, you specify a user name. You also can specify the following parameters:

- A password

- A management privilege level, which can be one of the following:
 - **Super User level** – Allows complete read-and-write access to the system. This is generally for system administrators and is the only privilege level that allows you to configure passwords. This is the default.
 - **Port Configuration level** – Allows read-and-write access for specific ports but not for global (system-wide) parameters.
 - **Read Only level** – Allows access to the Privileged EXEC mode and CONFIG mode but only with read access.

Configuring a local user account

To configure a local user account, enter a command such as the following at the global CONFIG level of the CLI.

```
ServerIronADX(config)# username wonka password willy
```

This command adds a local user account with the user name “wonka” and the password “willy”. This account has the Super User privilege level; this user has full access to all configuration and display features.

NOTE

If you configure local user accounts, you must grant Super User level access to at least one account before you add accounts with other privilege levels. You need the Super User account to make further administrative changes.

```
ServerIronADX(config)# username waldo privilege 5 password whereis
```

This command adds a user account for user name “waldo”, password “whereis”, with the Read Only privilege level. Waldo can look for information but cannot make configuration changes.

Syntax: **[no] username** *<user-string>* **privilege** *<privilege-level>* **password** | **nopassword** *<password-string>*

The **privilege** parameter specifies the privilege level for the account. You can specify one of the following:

- **0** – Super User level (full read-write access)
- **4** – Port Configuration level
- **5** – Read Only level

The default privilege level is **0**. If you want to assign Super User level access to the account, you can enter the command without **privilege 0**, as shown in the command example above.

The **password** | **nopassword** parameter indicates whether the user must enter a password. If you specify **password**, enter the string for the user's password.

NOTE

You must be logged on with Super User access (privilege level 0) to add user accounts or configure other access parameters.

To display user account information, enter the following command.

```
ServerIronADX(config)# show users
Username          Password          Encrypt   Priv Status   Expire
Time
=====
==
admin              $1$T62..hu1$hmRolcVlVwc.FCtXVD6h9/ enabled    0    enabled
Never
```

Syntax: show users

Configuring TACACS or TACACS+ security

You can use the security protocol Terminal Access Controller Access Control System (TACACS) or TACACS+ to authenticate the following kinds of access to the ServerIron.

- Telnet access
- SSH access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

NOTE

[You cannot authenticate IronView \(SNMP\) access to a ServerIron using TACACS or TACACS+.](#)

The TACACS and TACACS+ protocols define how authentication, authorization, and accounting information is sent between a ServerIron and an authentication database on a TACACS or TACACS+ server. TACACS or TACACS+ services are maintained in a database, typically on a UNIX workstation or PC with a TACACS or TACACS+ server running.

How TACACS+ differs from TACACS

TACACS is a simple UDP-based access control protocol originally developed by BBN for MILNET. TACACS+ is an enhancement to TACACS and uses TCP to ensure reliable delivery.

TACACS+ is an enhancement to the TACACS security protocol. TACACS+ improves on TACACS by separating the functions of authentication, authorization, and accounting (AAA) and by encrypting all traffic between the ServerIron and the TACACS+ server. TACACS+ allows for arbitrary length and content authentication exchanges, which allow any authentication mechanism to be utilized with the ServerIron. TACACS+ is extensible to provide for site customization and future development features. The protocol allows the ServerIron to request very precise access control and allows the TACACS+ server to respond to each component of that request.

NOTE

TACACS+ provides for authentication, authorization, and accounting, but an implementation or configuration is not required to employ all three.

TACACS or TACACS+ authentication, authorization, and accounting

When you configure a ServerIron to use a TACACS or TACACS+ server for authentication, the device prompts users who are trying to access the CLI for a user name and password, then verifies the password with the TACACS or TACACS+ server.

If you are using TACACS+, Brocade recommends that you also configure **authorization**, in which the ServerIron consults a TACACS+ server to determine which management privilege level (and which associated set of commands) an authenticated user is allowed to use. You can also optionally configure **accounting**, which causes the ServerIron to log information on the TACACS+ server when specified events occur on the device.

NOTE

By default, a user logging into the device through Telnet or SSH would first enter the User EXEC level. The user can enter the **enable** command to get to the Privileged EXEC level.

A user that is successfully authenticated can be automatically placed at the Privileged EXEC level after login. Refer to [“Entering privileged EXEC mode after a Telnet or SSH login”](#) on page 115.

TACACS authentication

When TACACS authentication takes place, the following events occur.

1. A user attempts to gain access to the ServerIron by doing one of the following:
 - Logging into the device using Telnet, SSH, or the Web Management Interface
 - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The ServerIron sends a request containing the username and password to the TACACS server.
5. The username and password are validated in the TACACS server's database.
6. If the password is valid, the user is authenticated.

TACACS+ authentication

When TACACS+ authentication takes place, the following events occur.

1. A user attempts to gain access to the ServerIron by doing one of the following:
 - Logging into the device using Telnet, SSH, or the Web Management Interface
 - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username.
3. The user enters a username.
4. The ServerIron obtains a password prompt from a TACACS+ server.
5. The user is prompted for a password.
6. The user enters a password.
7. The ServerIron sends the password to the TACACS+ server.
8. The password is validated in the TACACS+ server's database.

9. If the password is valid, the user is authenticated.

TACACS+ authorization

ServerIrons support two kinds of TACACS+ authorization:

- Exec authorization determines a user's privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

When TACACS+ exec authorization takes place, the following events occur.

1. A user logs into the ServerIron using Telnet, SSH, or the Web Management Interface
2. The user is authenticated.
3. The ServerIron consults the TACACS+ server to determine the privilege level of the user.
4. The TACACS+ server sends back a response containing an A-V (Attribute-Value) pair with the privilege level of the user.
5. The user is granted the specified privilege level.

When TACACS+ command authorization takes place, the following events occur.

1. A Telnet, SSH, or Web Management Interface user previously authenticated by a TACACS+ server enters a command on the ServerIron.
2. The ServerIron looks at its configuration to see if the command is at a privilege level that requires TACACS+ command authorization.
3. If the command belongs to a privilege level that requires authorization, the ServerIron consults the TACACS+ server to see if the user is authorized to use the command.
4. If the user is authorized to use the command, the command is executed.

TACACS+ accounting

TACACS+ accounting works as follows.

1. One of the following events occur on the ServerIron:
 - A user logs into the management interface using Telnet or SSH
 - A user enters a command for which accounting has been configured
 - A system event occurs, such as a reboot or reloading of the configuration file
2. The ServerIron checks its configuration to see if the event is one for which TACACS+ accounting is required.
3. If the event requires TACACS+ accounting, the ServerIron sends a TACACS+ Accounting Start packet to the TACACS+ accounting server, containing information about the event.
4. The TACACS+ accounting server acknowledges the Accounting Start packet.
5. The TACACS+ accounting server records information about the event.
6. When the event is concluded, the ServerIron sends an Accounting Stop packet to the TACACS+ accounting server.
7. The TACACS+ accounting server acknowledges the Accounting Stop packet.

AAA operations for TACACS or TACACS+

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a ServerIron that has TACACS or TACACS+ security configured.

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default <i><method-list></i> <hr/> Exec authorization (TACACS+): aaa authorization exec default tacacs+ <hr/> System accounting start (TACACS+): aaa accounting system default start-stop <i><method-list></i>
User logs in using Telnet or SSH	Login authentication: aaa authentication login default <i><method-list></i> <hr/> Exec authorization (TACACS+): aaa authorization exec default tacacs+ <hr/> Exec accounting start (TACACS+): aaa accounting exec default <i><method-list></i> System accounting start (TACACS+): aaa accounting system default start-stop <i><method-list></i>
User logs into the Web Management Interface	Web authentication: aaa authentication web-server default <i><method-list></i> <hr/> Exec authorization (TACACS+): aaa authorization exec default tacacs+
User logs out of Telnet or SSH session	Command accounting (TACACS+): aaa accounting commands <i><privilege-level></i> default start-stop <i><method-list></i> EXEC accounting stop (TACACS+): aaa accounting exec default start-stop <i><method-list></i>
User enters system commands (for example, reload , boot system)	Command authorization (TACACS+): aaa authorization commands <i><privilege-level></i> default <i><method-list></i> <hr/> Command accounting (TACACS+): aaa accounting commands <i><privilege-level></i> default start-stop <i><method-list></i> System accounting stop (TACACS+): aaa accounting system default start-stop <i><method-list></i>
User enters the command: [no] aaa accounting system default start-stop <i><method-list></i>	Command authorization (TACACS+): aaa authorization commands <i><privilege-level></i> default <i><method-list></i> <hr/> Command accounting (TACACS+): aaa accounting commands <i><privilege-level></i> default start-stop <i><method-list></i> System accounting start (TACACS+): aaa accounting system default start-stop <i><method-list></i>

User action	Applicable AAA operations
User enters other commands	Command authorization (TACACS+): aaa authorization commands <privilege-level> default <method-list>
	Command accounting (TACACS+): aaa accounting commands <privilege-level> default start-stop <method-list>

AAA security for commands pasted into the running-config

If AAA security is enabled on the device, commands pasted into the running-config are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running-config, and AAA command authorization or accounting is configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running-config. The server performing the AAA operations should be reachable when you paste the commands into the running-config file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

TACACS or TACACS+ configuration considerations

Consider the following:

- You must deploy at least one TACACS or TACACS+ server in your network.
- ServerIron supports authentication using up to eight TACACS or TACACS+ servers. The device tries to use the servers in the order you add them to the device's configuration.
- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select TACACS+ as the primary authentication method for Telnet CLI access, but you cannot also select RADIUS authentication as a primary method for the same type of access. However, you can configure backup authentication methods for each access type.
- You can configure the ServerIron to authenticate using a TACACS or TACACS+ server, not both.

TACACS configuration procedure

For TACACS configurations, use the following procedure.

1. Identify TACACS servers. Refer to ["Identifying the TACACS or TACACS+ servers"](#) on page 111.
2. Set optional parameters. Refer to ["Setting optional TACACS or TACACS+ parameters"](#) on page 112.
3. Configure authentication-method lists. Refer to ["Configuring authentication-method lists for TACACS or TACACS+"](#) on page 114.

TACACS+ configuration procedure

For TACACS+ configurations, use the following procedure.

1. Identify TACACS+ servers. Refer to [“Identifying the TACACS or TACACS+ servers”](#) on page 111.
2. Set optional parameters. Refer to [“Setting optional TACACS or TACACS+ parameters”](#) on page 112.
3. Configure authentication-method lists. Refer to [“Configuring authentication-method lists for TACACS or TACACS+”](#) on page 114.
4. Optionally configure TACACS+ authorization. Refer to [“Configuring TACACS+ authorization”](#) on page 116.
5. Optionally configure TACACS+ accounting. Refer to [“Configuring TACACS+ accounting”](#) on page 119.

Identifying the TACACS or TACACS+ servers

To use TACACS or TACACS+ servers to authenticate access to a ServerIron, you must identify the servers to the ServerIron device.

Example To identify three TACACS or TACACS+ servers

```
ServerIronADX(config)# tacacs-server host 207.94.6.161
ServerIronADX(config)# tacacs-server host 207.94.6.191
ServerIronADX(config)# tacacs-server host 207.94.6.122
```

Syntax: `tacacs-server <ip-addr>|<hostname> [auth-port <number>]`

The `<ip-addr>|<hostname>` parameter specifies the IP address or host name of the server. You can enter up to eight `tacacs-server host` commands to specify up to eight different servers.

NOTE

To specify the server's host name instead of its IP address, you must first identify a DNS server using the `ip dns server-address <ip-addr>` command at the global CONFIG level.

If you add multiple TACACS or TACACS+ authentication servers to the ServerIron, it tries to reach them in the order you add them. For example, if you add three servers in the following order, the software tries the servers in the same order.

1. 207.94.6.161
2. 207.94.6.191
3. 207.94.6.122

You can remove a TACACS or TACACS+ server by entering `no` followed by the `tacacs-server` command. For example, to remove 207.94.6.161, enter the following command.

```
ServerIronADX(config)# no tacacs-server host 207.94.6.161
```

NOTE

If you erase a `tacacs-server` command (by entering `no` followed by the command), make sure you also erase the `aaa` commands that specify TACACS or TACACS+ as an authentication method. (Refer to [“Configuring authentication-method lists for TACACS or TACACS+”](#) on page 114.) Otherwise, when you exit from the CONFIG mode or from a Telnet session, the system continues to believe it is TACACS or TACACS+ enabled and you will not be able to access the system.

The `auth-port` parameter specifies the UDP (for TACACS) or TCP (for TACACS+) port number of the authentication port on the server. The default port number is 49.

Specifying different servers for individual AAA functions

In a TACACS+ configuration, you can designate a server to handle a specific AAA task. For example, you can designate one TACACS+ server to handle authorization and another TACACS+ server to handle accounting. You can set the TACACS+ key for each server.

To specify different TACACS+ servers for authentication, authorization, and accounting, enter commands such as the following.

```
ServerIronADX(config)# tacacs-server host 1.2.3.4 auth-port 49
authentication-only key abc
ServerIronADX(config)# tacacs-server host 1.2.3.5 auth-port 49 authorization-only
key def
ServerIronADX(config)# tacacs-server host 1.2.3.6 auth-port 49 accounting-only
key ghi
```

Syntax: `tacacs-server host <ip-addr> | <server-name> [authentication-only | authorization-only | accounting-only | default] [key <string>]`

The **default** parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization or accounting. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until a server that can perform the requested function is found, or every server in the configured list has been tried.

Setting optional TACACS or TACACS+ parameters

You can set the following optional parameters in a TACACS or TACACS+ configuration:

- **TACACS+ key** – This parameter specifies the value that the ServerIron sends to the TACACS+ server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the ServerIron will resend an authentication request when the TACACS or TACACS+ server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- **Dead time** – This parameter specifies how long the ServerIron waits for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 – 5 seconds. The default is 3 seconds.
- **Timeout** – This parameter specifies how many seconds the ServerIron waits for a response from a TACACS or TACACS+ server before either retrying the authentication request, or determining that the TACACS or TACACS+ servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the TACACS+ key

NOTE

The **tacacs-server key** command applies only to TACACS+ servers, not to TACACS servers. If you are configuring TACACS, do not configure a key on the TACACS server and do not enter a key on the ServerIron.

To specify a TACACS+ server key, enter a command such as the following.


```
ServerIronADX(config)# tacacs-server key rk Wong
```

Syntax: `tacacs-server key [0 | 1] <string>`

The **key** parameter in the **tacacs-server** command is used to encrypt TACACS+ packets before they are sent over the network. The value for the **key** parameter on the ServerIron should match the one configured on the TACACS+ server. The **key** can be from 1 – 32 characters in length and cannot include any space characters.

When you display the configuration of the ServerIron, the TACACS+ keys are encrypted.

Example

```
ServerIronADX(config)# tacacs-server key 1 abc
ServerIronADX(config)# write terminal
...
tacacs-server host 1.2.3.5 auth-port 49
tacacs key 1 $!2d
```

NOTE

Encryption of the TACACS+ keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the retransmission limit

To set the TACACS or TACACS+ retransmit limit, enter a command such as the following.

```
ServerIronADX(config)# tacacs-server retransmit 5
```

Syntax: `tacacs-server retransmit <number>`

The **retransmit** parameter specifies how many times the ServerIron will resend an authentication request when the TACACS or TACACS+ server does not respond. The retransmit limit can be from 0 – 5 times. The default is 3 times.

Setting the dead time parameter

To set the TACACS or TACACS+ dead-time value, enter a command such as the following.

```
ServerIronADX(config)# tacacs-server dead-time 5
```

Syntax: `tacacs-server dead-time <number>`

The **dead-time** parameter specifies how long the ServerIron waits for the primary authentication server to reply before deciding the server is dead and trying to authenticate using the next server. The dead-time value can be from 1 – 5 seconds. The default is 3 seconds.

Setting the timeout parameter

```
ServerIronADX(config)# tacacs-server timeout 5
```

Syntax: `tacacs-server timeout <number>`

The **timeout** parameter specifies how many seconds the ServerIron waits for a response from the TACACS or TACACS+ server before either retrying the authentication request, or determining that the TACACS or TACACS+ server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Configuring authentication-method lists for TACACS or TACACS+

You can use TACACS or TACACS+ to authenticate Telnet or SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring TACACS or TACACS+ authentication, you create authentication-method lists specifically for these access methods, specifying TACACS or TACACS+ as the primary authentication method.

Within the authentication-method list, TACACS or TACACS+ is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If TACACS or TACACS+ authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list.

When you configure authentication-method lists for TACACS or TACACS+ authentication, you must create a separate authentication-method list for Telnet or SSH CLI access, and for access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication-method list that specifies TACACS or TACACS+ as the primary authentication method for securing Telnet or SSH access to the CLI.

```
ServerIronADX(config)# enable telnet authentication
ServerIronADX(config)# aaa authentication login default tacacs local
```

The commands above cause TACACS or TACACS+ to be the primary authentication method for securing Telnet or SSH access to the CLI. If TACACS or TACACS+ authentication fails due to an error with the server, authentication is performed using local user accounts instead.

To create an authentication-method list that specifies TACACS or TACACS+ as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI.

```
ServerIronADX(config)# aaa authentication enable default tacacs local none
```

The command above causes TACACS or TACACS+ to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If TACACS or TACACS+ authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

Syntax: `[no] aaa authentication enable | login default <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]`

The **web-server | enable | login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE

If you configure authentication for Web management access, authentication is performed each time a page is requested from the server. When frames are enabled on the Web Management Interface, the browser sends an HTTP request for each frame. The ServerIron authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web Management Interface.

The *<method1>* parameter specifies the primary authentication method. The remaining optional *<method>* parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

TABLE 7 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the enable telnet password... command. Refer to “Setting a Telnet password” on page 100.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the enable super-user-password... command. Refer to “Setting passwords for management privilege levels” on page 100.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the username... command. Refer to “Configuring a local user account” on page 105.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the tacacs-server command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the tacacs-server command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the radius-server command.
none	Do not use any authentication method. The device automatically permits access.

NOTE

For examples of how to define authentication-method lists for types of authentication other than TACACS or TACACS+, refer to [“Configuring authentication-method lists”](#) on page 136.

Entering privileged EXEC mode after a Telnet or SSH login

By default, a user enters User EXEC mode after a successful login through Telnet or SSH. Optionally, you can configure the device so that a user enters Privileged EXEC mode after a Telnet or SSH login. To do this, use the following command.

```
ServerIronADX(config)# aaa authentication login privilege-mode
```

Syntax: **aaa authentication login privilege-mode**

The user's privilege level is based on the privilege level granted during login.

Configuring enable authentication to prompt for password only

If Enable authentication is configured on the device, when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI, by default he or she is prompted for a username and password. You can configure the ServerIron device to prompt only for a password. The device uses the username entered at login, if one is available. If no username was entered at login, the device prompts for both username and password.

To configure the ServerIron to prompt only for a password when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI.

```
ServerIronADX(config)# aaa authentication enable implicit-user
```

Syntax: **[no] aaa authentication enable implicit-user**

Telnet or SSH prompts when TACACS+ server is unavailable

When TACACS+ is the first method in the authentication method list, the device displays the login prompt received from the TACACS+ server. If a user attempts to login through Telnet or SSH, but none of the configured TACACS+ servers are available, the following takes place:

- If the next method in the authentication method list is "enable", the login prompt is skipped, and the user is prompted for the Enable password (that is, the password configured with the **enable super-user-password** command).
- If the next method in the authentication method list is "line", the login prompt is skipped, and the user is prompted for the Line password (that is, the password configured with the **enable telnet password** command).

Configuring TACACS+ authorization

ServerIron supports TACACS+ authorization for controlling access to management functions in the CLI. Two kinds of TACACS+ authorization are supported:

- Exec authorization determines a user's privilege level when they are authenticated
- Command authorization consults a TACACS+ server to get authorization for commands entered by the user

Configuring Exec authorization

When TACACS+ exec authorization is performed, the ServerIron consults a TACACS+ server to determine the privilege level of the authenticated user. To configure TACACS+ exec authorization on the ServerIron, enter the following command.

```
ServerIronADX(config)# aaa authorization exec default tacacs+
```

Syntax: `aaa authorization exec default tacacs+ | none`

If you specify **none**, or omit the **aaa authorization exec** command from the device's configuration, no exec authorization is performed.

A user's privilege level is obtained from the TACACS+ server in the "brocade-privlvl" A-V pair. If the **aaa authorization exec default tacacs** command exists in the configuration, the device assigns the user the privilege level specified by this A-V pair. If the command does not exist in the configuration, then the value in the "brocade-privlvl" A-V pair is ignored, and the user is granted Super User access.

NOTE

If the **aaa authorization exec default tacacs+** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the "brocade-privlvl" A-V pair received from the TACACS+ server. If the **aaa authorization exec default tacacs+** command does not exist in the configuration, then the value in the "brocade-privlvl" A-V pair is ignored, and the user is granted Super User access.

Also note that in order for the **aaa authorization exec default tacacs+** command to work, either the **aaa authentication enable default tacacs+** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring an attribute-value pair on the TACACS+ server

During TACACS+ exec authorization, the ServerIron expects the TACACS+ server to send a response containing an A-V (Attribute-Value) pair that specifies the privilege level of the user. When the ServerIron receives the response, it extracts an A-V pair configured for the Exec service and uses it to determine the user's privilege level.

To set a user's privilege level, you can configure the "brocade-privlvl" A-V pair for the Exec service on the TACACS+ server.

Example

```
user=bob {
    default service = permit
    member admin
    # Global password
    global = cleartext "cat"
    service = exec {
        brocade-privlvl = 0
    }
}
```

In this example, the A-V pair `brocade-privlvl = 0` grants the user full read-write access. The value in the `brocade-privlvl` A-V pair is an integer that indicates the privilege level of the user. Possible values are 0 for super-user level, 4 for port-config level, or 5 for read-only level. If a value other than 0, 4, or 5 is specified in the `brocade-privlvl` A-V pair, the default privilege level of 5 (read-only) is used. The `brocade-privlvl` A-V pair can also be embedded in the group configuration for the user. Refer to your TACACS+ documentation for the configuration syntax relevant to your server.

If the `brocade-privlvl` A-V pair is not present, the ServerIron extracts the last A-V pair configured for the Exec service that has a numeric value. The ServerIron uses this A-V pair to determine the user's privilege level.

Example

```
user=bob {
    default service = permit
    member admin
    # Global password
    global = cleartext "cat"
    service = exec {
        privlvl = 15
    }
}
```

The attribute name in the A-V pair is not significant; the ServerIron uses the last one that has a numeric value. However, the ServerIron interprets the value for a non-`brocade-privlvl` A-V pair differently than it does for a `brocade-privlvl` A-V pair. The following table lists how the ServerIron associates a value from a non-`brocade-privlvl` A-V pair with a Brocade privilege level.

TABLE 8 Brocade equivalents for non-`brocade-privlvl` A-V pair values

Value for non- <code>brocade-privlvl</code> A-V Pair	Brocade privilege level
15	0 (super-user)
From 14 – 1	4 (port-config)
Any other number or 0	5 (read-only)

In the example above, the A-V pair configured for the Exec service is `privlvl = 15`. The ServerIron uses the value in this A-V pair to set the user's privilege level to 0 (super-user), granting the user full read-write access.

In a configuration that has both a "`brocade-privlvl`" A-V pair and a non-"`brocade-privlvl`" A-V pair for the Exec service, the non-"`brocade-privlvl`" A-V pair is ignored.

Example

```
user=bob {
    default service = permit
    member admin
    # Global password
    global = cleartext "cat"
    service = exec {
        brocade-privlvl = 4
        privlvl = 15
    }
}
```

In this example, the user would be granted a privilege level of 4 (port-config level). The `privlvl = 15` A-V pair is ignored by the ServerIron.

If the TACACS+ server has no A-V pair configured for the Exec service, the default privilege level of 5 (read-only) is used.

Configuring command authorization

When TACACS+ command authorization is enabled, the ServerIron consults a TACACS+ server to get authorization for commands entered by the user.

You enable TACACS+ command authorization by specifying a privilege level whose commands require authorization. For example, to configure the ServerIron to perform authorization for the commands available at the Super User privilege level (that is, all commands on the device), enter the following command.

```
ServerIronADX(config)# aaa authorization commands 0 default tacacs+
```

Syntax: `aaa authorization commands <privilege-level> default tacacs+ | radius | none`

The `<privilege-level>` parameter can be one of the following:

- **0** – Authorization is performed for commands available at the Super User level (all commands)
- **4** – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE

TACACS+ command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web Management Interface or [IronView](#).

TACACS+ command authorization is not performed for the following commands:

- At all levels: **exit**, **logout**, **end**, and **quit**.
- At the Privileged EXEC level: **enable** or **enable <text>**, where `<text>` is the password configured for the Super User privilege level.

If configured, command accounting is performed for these commands.

Command authorization and accounting for console commands

The ServerIron supports command authorization and command accounting for CLI commands entered at the console. To configure the device to perform command authorization and command accounting for console commands, enter the following.

```
ServerIronADX(config)# enable aaa console
```

Syntax: enable aaa console

Configuring TACACS+ accounting

ServerIron supports TACACS+ accounting for recording information about user activity and system events. When you configure TACACS+ accounting on a ServerIron, information is sent to a TACACS+ accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring TACACS+ accounting for Telnet or SSH (shell) access

To send an Accounting Start packet to the TACACS+ accounting server when an authenticated user establishes a Telnet or SSH session on the ServerIron, and an Accounting Stop packet when the user logs out.

```
ServerIronADX(config)# aaa accounting exec default start-stop tacacs+
```

Syntax: aaa accounting exec default start-stop radius | tacacs+ | none

Configuring TACACS+ accounting for CLI commands

You can configure TACACS+ accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure the ServerIron to perform TACACS+ accounting for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
ServerIronADX(config)# aaa accounting commands 0 default start-stop tacacs+
```

An Accounting Start packet is sent to the TACACS+ accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

NOTE

If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: aaa accounting commands <privilege-level> default start-stop radius | tacacs+ | none

The <privilege-level> parameter can be one of the following:

- **0** – Records commands available at the Super User level (all commands)
- **4** – Records commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Records commands available at the Read Only level (read-only commands)

Configuring TACACS+ accounting for system events

You can configure TACACS+ accounting to record when system events occur on the ServerIron. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the TACACS+ accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed.

```
ServerIronADX(config)# aaa accounting system default start-stop tacacs+
```

Syntax: `aaa accounting system default start-stop radius | tacacs+ | none`

Configuring an interface as the source for all TACACS or TACACS+ packets

You can designate the lowest-numbered IP address configured on an Ethernet port, [POS port](#), loopback interface, or virtual interface as the source IP address for all TACACS or TACACS+ packets from the Layer 3 Switch. Identifying a single source IP address for TACACS or TACACS+ packets provides the following benefits:

- If your TACACS or TACACS+ server is configured to accept packets only from specific links or IP addresses, you can use this feature to simplify configuration of the TACACS or TACACS+ server by configuring the ServerIron to always send the TACACS or TACACS+ packets from the same link or source address.
- If you specify a loopback interface as the single source for TACACS or TACACS+ packets, TACACS or TACACS+ servers can receive the packets regardless of the states of individual links. Thus, if a link to the TACACS or TACACS+ server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, TACACS or TACACS+, and RADIUS packets. You can configure a source interface for one or more of these types of packets.

To specify an Ethernet [or POS](#) port or a loopback or virtual interface as the source for all TACACS or TACACS+ packets from the device, use the following CLI method. The software uses the lowest-numbered IP address configured on the port or interface as the source IP address for TACACS or TACACS+ packets originated by the device.

To specify the lowest-numbered IP address configured on a virtual interface as the device's source for all TACACS or TACACS+ packets, enter commands such as the following.

```
ServerIronADX(config)# interface ve 1
ServerIronADX(config-vif-1)# ip address 10.0.0.3/24
ServerIronADX(config-vif-1)# exit
ServerIronADX(config)# ip tacacs source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all TACACS or TACACS+ packets from the Layer 3 Switch.

Syntax: `ip tacacs source-interface ethernet <portnum> | pos <portnum> | loopback <num> | ve <num>`

The `<num>` parameter is a loopback interface or virtual interface number. If you specify an Ethernet or POS port, the `<portnum>` is the port's number (including the slot number, if you are configuring a chassis device).

Displaying TACACS or TACACS+ statistics and configuration information

The **show aaa** command displays information about all TACACS+ and RADIUS servers identified on the device.

Example

```
ServerIronADX# show aaa
Tacacs+ key: brocade
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                  opens=6 closes=3 timeouts=3 errors=0
                  packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server: 207.95.6.90 Auth Port=1645 Acct Port=1646:
                  opens=2 closes=1 timeouts=1 errors=0
                  packets in=1 packets out=4
no connection
```

The following table describes the TACACS or TACACS+ information displayed by the **show aaa** command.

TABLE 9 Output of the show aaa command for TACACS or TACACS+

Field	Description
Tacacs+ key	The setting configured with the tacacs-server key command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (...) is displayed instead of the text.
Tacacs+ retries	The setting configured with the tacacs-server retransmit command.
Tacacs+ timeout	The setting configured with the tacacs-server timeout command.
Tacacs+ dead-time	The setting configured with the tacacs-server dead-time command.
Tacacs+ Server	For each TACACS or TACACS+ server, the IP address, port, and the following statistics are displayed: opensNumber of times the port was opened for communication with the server closesNumber of times the port was closed normally timeoutsNumber of times port was closed due to a timeout errorsNumber of times an error occurred while opening the port packets inNumber of packets received from the server packets outNumber of packets sent to the server
connection	The current connection status. This can be "no connection" or "connection active".

The **show web** command displays the privilege level of Web Management Interface users.

Example

```
ServerIronADX(config)# show web
User                               Privilege    IP address
set                                0            192.168.1.234
```

Syntax: show web

Configuring RADIUS security

You can use a Remote Authentication Dial In User Service (RADIUS) server to secure the following types of access to the Brocade [Layer 2 Switch](#) or Layer 3 Switch:

- Telnet access
- SSH access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

NOTE

ServerIrons do not support RADIUS security for SNMP ([IronView](#)) access.

RADIUS authentication, authorization, and accounting

When RADIUS *authentication* is implemented, the ServerIron consults a RADIUS server to verify user names and passwords. You can optionally configure RADIUS *authorization*, in which the ServerIron consults a list of commands supplied by the RADIUS server to determine whether a user can execute a command he or she has entered, as well as *accounting*, which causes the ServerIron to log information on a RADIUS accounting server when specified events occur on the device.

NOTE

By default, a user logging into the device through Telnet or SSH first enters the User EXEC level. The user can then enter the **enable** command to get to the Privileged EXEC level.

A user that is successfully authenticated can be automatically placed at the Privileged EXEC level after login. Refer to [“Entering privileged EXEC mode after a Telnet or SSH login”](#) on page 131.

RADIUS authentication

When RADIUS authentication takes place, the following events occur.

1. A user attempts to gain access to the ServerIron by doing one of the following:
 - Logging into the device using Telnet, SSH, or the Web Management Interface
 - Entering the Privileged EXEC level or CONFIG level of the CLI
2. The user is prompted for a username and password.
3. The user enters a username and password.
4. The ServerIron sends a RADIUS Access-Request packet containing the username and password to the RADIUS server.
5. The RADIUS server validates the ServerIron using a shared secret (the RADIUS key).
6. The RADIUS server looks up the username in its database.

7. If the username is found in the database, the RADIUS server validates the password.
8. If the password is valid, the RADIUS server sends an Access-Accept packet to the ServerIron, authenticating the user. Within the Access-Accept packet are three Brocade vendor-specific attributes that indicate:
 - The privilege level of the user
 - A list of commands
 - Whether the user is allowed or denied usage of the commands in the list

The last two attributes are used with RADIUS authorization, if configured.

9. The user is authenticated, and the information supplied in the Access-Accept packet for the user is stored on the ServerIron. The user is granted the specified privilege level. If you configure RADIUS authorization, the user is allowed or denied usage of the commands in the list.

RADIUS authorization

When RADIUS authorization takes place, the following events occur.

1. A user previously authenticated by a RADIUS server enters a command on the ServerIron.
2. The ServerIron looks at its configuration to see if the command is at a privilege level that requires RADIUS command authorization.
3. If the command belongs to a privilege level that requires authorization, the ServerIron looks at the list of commands delivered to it in the RADIUS Access-Accept packet when the user was authenticated. (Along with the command list, an attribute was sent that specifies whether the user is permitted or denied usage of the commands in the list.)

NOTE

After RADIUS authentication takes place, the command list resides on the ServerIron. The RADIUS server is not consulted again once the user has been authenticated. This means that any changes made to the user's command list on the RADIUS server are not reflected until the next time the user is authenticated by the RADIUS server, and the new command list is sent to the ServerIron.

4. If the command list indicates that the user is authorized to use the command, the command is executed.

RADIUS accounting

RADIUS accounting works as follows.

1. One of the following events occur on the ServerIron:
 - A user logs into the management interface using Telnet or SSH
 - A user enters a command for which accounting has been configured
 - A system event occurs, such as a reboot or reloading of the configuration file
2. The ServerIron checks its configuration to see if the event is one for which RADIUS accounting is required.
3. If the event requires RADIUS accounting, the ServerIron sends a RADIUS Accounting Start packet to the RADIUS accounting server, containing information about the event.

4. The RADIUS accounting server acknowledges the Accounting Start packet.
5. The RADIUS accounting server records information about the event.
6. When the event is concluded, the ServerIron sends an Accounting Stop packet to the RADIUS accounting server.
7. The RADIUS accounting server acknowledges the Accounting Stop packet.

AAA operations for RADIUS

The following table lists the sequence of authentication, authorization, and accounting operations that take place when a user gains access to a ServerIron that has RADIUS security configured.

User action	Applicable AAA operations
User attempts to gain access to the Privileged EXEC and CONFIG levels of the CLI	Enable authentication: aaa authentication enable default <method-list> <hr/> System accounting start: aaa accounting system default start-stop <method-list>
User logs in using Telnet or SSH	Login authentication: aaa authentication login default <method-list> <hr/> EXEC accounting Start: aaa accounting exec default start-stop <method-list> System accounting Start: aaa accounting system default start-stop <method-list>
User logs into the Web Management Interface	Web authentication: aaa authentication web-server default <method-list>
User logs out of Telnet or SSH session	Command authorization for logout command: aaa authorization commands <privilege-level> default <method-list> <hr/> Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list> EXEC accounting stop: aaa accounting exec default start-stop <method-list>
User enters system commands (for example, reload , boot system)	Command authorization: aaa authorization commands <privilege-level> default <method-list> <hr/> Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list> System accounting stop: aaa accounting system default start-stop <method-list>
User enters the command: [no] aaa accounting system default start-stop <method-list>	Command authorization: aaa authorization commands <privilege-level> default <method-list> <hr/> Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list> System accounting start: aaa accounting system default start-stop <method-list>

User action	Applicable AAA operations
User enters other commands	Command authorization: aaa authorization commands <privilege-level> default <method-list>
	Command accounting: aaa accounting commands <privilege-level> default start-stop <method-list>

AAA security for commands pasted into the running-config

If AAA security is enabled on the device, commands pasted into the running-config are subject to the same AAA operations as if they were entered manually.

When you paste commands into the running-config, and AAA command authorization or accounting is configured on the device, AAA operations are performed on the pasted commands. The AAA operations are performed before the commands are actually added to the running-config. The server performing the AAA operations should be reachable when you paste the commands into the running-config file. If the device determines that a pasted command is invalid, AAA operations are halted on the remaining commands. The remaining commands may not be executed if command authorization is configured.

NOTE

Since RADIUS command authorization relies on a list of commands received from the RADIUS server when authentication is performed, it is important that you use RADIUS authentication when you also use RADIUS command authorization.

RADIUS NAS-Identifier

RADIUS NAS-Identifier gives customers using multi-vendor networks identifiers for ServerIron so their RADIUS servers can send the correct VSAs to the device. Customers who use multi-vendor networks require a default value for ServerIron and the ability to configure one string per device for different business and operational functions.

The ServerIron RADIUS implementation sends out a NAS-ID string in the access-request packets. To configure this feature use the following command:

```
ServerIronADX(config)#radius nas-identifier <string>
```

Syntax: radius nas-identifier <string>

- <string>—1 to 64 character string that identifies the NAS originating the access request. It is only used in access-request packets. Either NAS-IP-Address or NAS-Identifier must be present in an access-request packet.

RADIUS configuration considerations

Consider the following:

- You must deploy at least one RADIUS server in your network.
- ServerIron supports authentication using up to eight RADIUS servers. The device tries to use the servers in the order you add them to the device's configuration. If one RADIUS server is not responding, the ServerIron tries the next one in the list.

- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select RADIUS as the primary authentication method for Telnet CLI access, but you cannot also select TACACS+ authentication as the primary method for the same type of access. However, you can configure backup authentication methods for each access type.

RADIUS configuration procedure

Use the following procedure to configure a ServerIron for RADIUS.

1. Configure Brocade vendor-specific attributes on the RADIUS server. Refer to [“Configuring Brocade-specific attributes on the RADIUS server”](#) on page 126.
2. Identify the RADIUS server to the ServerIron. Refer to [“Identifying the RADIUS server to the ServerIron”](#) on page 127.
3. Set RADIUS parameters. Refer to [“Setting RADIUS parameters”](#) on page 128.
4. Configure authentication-method lists. Refer to [“Configuring authentication-method lists for RADIUS”](#) on page 129.
5. Optionally configure RADIUS authorization. Refer to [“Configuring RADIUS authorization”](#) on page 131.
6. Optionally configure RADIUS accounting. [“Configuring RADIUS accounting”](#) on page 133.

Configuring Brocade-specific attributes on the RADIUS server

During the RADIUS authentication process, if a user supplies a valid username and password, the RADIUS server sends an Access-Accept packet to the ServerIron, authenticating the user. Within the Access-Accept packet are three Brocade vendor-specific attributes that indicate:

- The privilege level of the user
- A list of commands
- Whether the user is allowed or denied usage of the commands in the list

You must add these three Brocade vendor-specific attributes to your RADIUS server’s configuration, and configure the attributes in the individual or group profiles of the users that will access the ServerIron.

Brocade’s Vendor-ID is **1991**, with Vendor-Type 1. The following table describes the Brocade vendor-specific attributes.

TABLE 10 Brocade vendor-specific attributes for RADIUS

Attribute name	Attribute ID	Data type	Description
<code>brocade-privilege-level</code>	1	integer	Specifies the privilege level for the user. This attribute can be set to one of the following: 0 Super User level – Allows complete read-and-write access to the system. This is generally for system administrators and is the only management privilege level that allows you to configure passwords. 4 Port Configuration level – Allows read-and-write access for specific ports but not for global (system-wide) parameters. 5 Read Only level – Allows access to the Privileged EXEC mode and CONFIG mode of the CLI but only with read access.
<code>brocade-command-string</code>	2	string	Specifies a list of CLI commands that are permitted or denied to the user when RADIUS authorization is configured. The commands are delimited by semi-colons (;). You can specify an asterisk (*) as a wildcard at the end of a command string. For example, the following command list specifies all show and debug ip commands, as well as the write terminal command: <code>show *; debug ip *; write term*</code>
<code>brocade-command-exception-flag</code>	3	integer	Specifies whether the commands indicated by the <code>brocade-command-string</code> attribute are permitted or denied to the user. This attribute can be set to one of the following: 0 Permit execution of the commands indicated by <code>brocade-command-string</code> , deny all other commands. 1 Deny execution of the commands indicated by <code>brocade-command-string</code> , permit all other commands.

Identifying the RADIUS server to the ServerIron

To use a RADIUS server to authenticate access to a ServerIron, you must identify the server to the ServerIron.

Example

```
ServerIronADX(config)# radius-server host 209.157.22.99
```

Syntax: `radius-server host <ip-addr> | <server-name> [auth-port <number> acct-port <number>]`

The `host <ip-addr> | <server-name>` parameter is either an IP address or an ASCII text string.

The `<auth-port>` parameter is the Authentication port number; it is an optional parameter. The default is 1645.

The `<acct-port>` parameter is the Accounting port number; it is an optional parameter. The default is 1646.

Specifying different servers for individual AAA functions

In a RADIUS configuration, you can designate a server to handle a specific AAA task. For example, you can designate one RADIUS server to handle authorization and another RADIUS server to handle accounting. You can specify individual servers for authentication and accounting, but not for authorization. You can set the RADIUS key for each server.

To specify different RADIUS servers for authentication, authorization, and accounting.

```
ServerIronADX(config)# radius-server host 1.2.3.4 authentication-only key abc
ServerIronADX(config)# radius-server host 1.2.3.5 authorization-only key def
ServerIronADX(config)# radius-server host 1.2.3.6 accounting-only key ghi
```

Syntax: `radius-server host <ip-addr> / <server-name> [authentication-only | accounting-only | default] [key 0 | 1 <string>]`

The **default** parameter causes the server to be used for all AAA functions.

After authentication takes place, the server that performed the authentication is used for authorization or accounting. If the authenticating server cannot perform the requested function, then the next server in the configured list of servers is tried; this process repeats until a server that can perform the requested function is found, or every server in the configured list has been tried.

Setting RADIUS parameters

You can set the following parameters in a RADIUS configuration:

- **RADIUS key** – This parameter specifies the value that the ServerIron sends to the RADIUS server when trying to authenticate user access.
- **Retransmit interval** – This parameter specifies how many times the ServerIron will resend an authentication request when the RADIUS server does not respond. The retransmit value can be from 1 – 5 times. The default is 3 times.
- **Timeout** – This parameter specifies how many seconds the ServerIron waits for a response from a RADIUS server before either retrying the authentication request, or determining that the RADIUS servers are unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

Setting the RADIUS key

The **key** parameter in the **radius-server** command is used to encrypt RADIUS packets before they are sent over the network. The value for the **key** parameter on the ServerIron should match the one configured on the RADIUS server. The key can be from 1 – 32 characters in length and cannot include any space characters.

To specify a RADIUS server key.

```
ServerIronADX(config)# radius-server key mirabeau
```

Syntax: `radius-server key [0 | 1] <string>`

When you display the configuration of the ServerIron, the RADIUS key is encrypted.

Example

```
ServerIronADX(config)# radius-server key 1 abc
ServerIronADX(config)# write terminal
...
radius-server host 1.2.3.5
radius key 1 $!2d
```

NOTE

Encryption of the RADIUS keys is done by default. The **0** parameter disables encryption. The **1** parameter is not required; it is provided for backwards compatibility.

Setting the retransmission limit

The **retransmit** parameter specifies the maximum number of retransmission attempts. When an authentication request times out, the Brocade software will retransmit the request up to the maximum number of retransmissions configured. The default retransmit value is 3 retries. The range of retransmit values is from 0–5.

To set the RADIUS retransmit limit.

```
ServerIronADX(config)# radius-server retransmit 5
```

Syntax: **radius-server retransmit** *<number>*

Setting the timeout parameter

The **timeout** parameter specifies how many seconds the ServerIron waits for a response from the RADIUS server before either retrying the authentication request, or determining that the RADIUS server is unavailable and moving on to the next authentication method in the authentication-method list. The timeout can be from 1 – 15 seconds. The default is 3 seconds.

```
ServerIronADX(config)# radius-server timeout 5
```

Syntax: **radius-server timeout** *<number>*

Configuring authentication-method lists for RADIUS

You can use RADIUS to authenticate Telnet or SSH access and access to Privileged EXEC level and CONFIG levels of the CLI. When configuring RADIUS authentication, you create authentication-method lists specifically for these access methods, specifying RADIUS as the primary authentication method.

Within the authentication-method list, RADIUS is specified as the primary authentication method and up to six backup authentication methods are specified as alternates. If RADIUS authentication fails due to an error, the device tries the backup authentication methods in the order they appear in the list.

When you configure authentication-method lists for RADIUS, you must create a separate authentication-method list for Telnet or SSH CLI access and for CLI access to the Privileged EXEC level and CONFIG levels of the CLI.

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing Telnet access to the CLI.

```
ServerIronADX(config)# enable telnet authentication
ServerIronADX(config)# aaa authentication login default radius local
```

The commands above cause RADIUS to be the primary authentication method for securing Telnet access to the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead.

To create an authentication-method list that specifies RADIUS as the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI.

```
ServerIronADX(config)# aaa authentication enable default radius local none
```

The command above causes RADIUS to be the primary authentication method for securing access to Privileged EXEC level and CONFIG levels of the CLI. If RADIUS authentication fails due to an error with the server, local authentication is used instead. If local authentication fails, no authentication is used; the device automatically permits access.

Syntax: `[no] aaa authentication enable | login default <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]`

The **web-server | enable | login** parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE

If you configure authentication for Web management access, authentication is performed each time a page is requested from the server. When frames are enabled on the Web Management Interface, the browser sends an HTTP request for each frame. The ServerIron authenticates each HTTP request from the browser. To limit authentications to one per page, disable frames on the Web Management Interface.

The *<method1>* parameter specifies the primary authentication method. The remaining optional *<method>* parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

TABLE 11 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the enable telnet password... command. Refer to “Setting a Telnet password” on page 100.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the enable super-user-password... command. Refer to “Setting passwords for management privilege levels” on page 100.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the username... command. Refer to “Configuring a local user account” on page 105.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the tacacs-server command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the tacacs-server command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the radius-server command.
none	Do not use any authentication method. The device automatically permits access.

NOTE

For examples of how to define authentication-method lists for types of authentication other than RADIUS, refer to [“Configuring authentication-method lists”](#) on page 136.

Entering privileged EXEC mode after a Telnet or SSH login

By default, a user enters User EXEC mode after a successful login through Telnet or SSH. Optionally, you can configure the device so that a user enters Privileged EXEC mode after a Telnet or SSH login. To do this, use the following command.

```
ServerIronADX(config)# aaa authentication login privilege-mode
```

Syntax: `aaa authentication login privilege-mode`

The user's privilege level is based on the privilege level granted during login.

Configuring enable authentication to prompt for password only

If Enable authentication is configured on the device, when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI, by default he or she is prompted for a username and password. You can configure the ServerIron to prompt only for a password. The device uses the username entered at login, if one is available. If no username was entered at login, the device prompts for both username and password.

To configure the ServerIron to prompt only for a password when a user attempts to gain Super User access to the Privileged EXEC and CONFIG levels of the CLI.

```
ServerIronADX(config)# aaa authentication enable implicit-user
```

Syntax: `[no] aaa authentication enable implicit-user`

Configuring RADIUS authorization

ServerIrons support RADIUS authorization for controlling access to management functions in the CLI. Two kinds of RADIUS authorization are supported:

- Exec authorization determines a user's privilege level when they are authenticated
- Command authorization consults a RADIUS server to get authorization for commands entered by the user

Configuring Exec authorization

When RADIUS exec authorization is performed, the ServerIron consults a RADIUS server to determine the privilege level of the authenticated user. To configure RADIUS exec authorization on the ServerIron, enter the following command.

```
ServerIronADX(config)# aaa authorization exec default radius
```

Syntax: `aaa authorization exec default radius | none`

If you specify **none**, or omit the **aaa authorization exec** command from the device's configuration, no exec authorization is performed.

NOTE

If the **aaa authorization exec default radius** command exists in the configuration, following successful authentication the device assigns the user the privilege level specified by the **brocade-privilege-level** attribute received from the RADIUS server. If the **aaa authorization exec default radius** command does not exist in the configuration, then the value in the **brocade-privilege-level** attribute is ignored, and the user is granted Super User access.

Also note that in order for the **aaa authorization exec default radius** command to work, either the **aaa authentication enable default radius** command, or the **aaa authentication login privilege-mode** command must also exist in the configuration.

Configuring command authorization

When RADIUS command authorization is enabled, the ServerIron consults the list of commands supplied by the RADIUS server during authentication to determine whether a user can execute a command he or she has entered.

You enable RADIUS command authorization by specifying a privilege level whose commands require authorization. For example, to configure the ServerIron to perform authorization for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
ServerIronADX(config)# aaa authorization commands 0 default radius
```

Syntax: **aaa authorization commands** *<privilege-level>* **default radius | tacacs+ | none**

The *<privilege-level>* parameter can be one of the following:

- **0** – Authorization is performed (that is, the ServerIron looks at the command list) for commands available at the Super User level (all commands)
- **4** – Authorization is performed for commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Authorization is performed for commands available at the Read Only level (read-only commands)

NOTE

RADIUS command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web Management Interface or [IronView](#).

NOTE

Since RADIUS command authorization relies on the command list supplied by the RADIUS server during authentication, you cannot perform RADIUS authorization without RADIUS authentication.

Command authorization and accounting for console commands

The ServerIron supports command authorization and command accounting for CLI commands entered at the console. To configure the device to perform command authorization and command accounting for console commands, enter the following.

```
ServerIronADX(config)# enable aaa console
```

Syntax: enable aaa console



DANGER

If you have previously configured the device to perform command authorization using a RADIUS server, entering the enable aaa console command may prevent the execution of any subsequent commands entered on the console.

NOTE

This happens because RADIUS command authorization requires a list of allowable commands from the RADIUS server. This list is obtained during RADIUS authentication. For console sessions, RADIUS authentication is performed only if you have configured Enable authentication and specified RADIUS as the authentication method (for example, with the **aaa authentication enable default radius** command). If RADIUS authentication is never performed, the list of allowable commands is never obtained from the RADIUS server. Consequently, there would be no allowable commands on the console.

Configuring RADIUS accounting

ServerIron supports RADIUS accounting for recording information about user activity and system events. When you configure RADIUS accounting on a ServerIron, information is sent to a RADIUS accounting server when specified events occur, such as when a user logs into the device or the system is rebooted.

Configuring RADIUS accounting for Telnet or SSH (Shell) access

To send an Accounting Start packet to the RADIUS accounting server when an authenticated user establishes a Telnet or SSH session on the ServerIron, and an Accounting Stop packet when the user logs out.

```
ServerIronADX(config)# aaa accounting exec default start-stop radius
```

Syntax: aaa accounting exec default start-stop radius | tacacs+ | none

Configuring RADIUS accounting for CLI commands

You can configure RADIUS accounting for CLI commands by specifying a privilege level whose commands require accounting. For example, to configure the ServerIron to perform RADIUS accounting for the commands available at the Super User privilege level (that is; all commands on the device), enter the following command.

```
ServerIronADX(config)# aaa accounting commands 0 default start-stop radius
```

An Accounting Start packet is sent to the RADIUS accounting server when a user enters a command, and an Accounting Stop packet is sent when the service provided by the command is completed.

NOTE

If authorization is enabled, and the command requires authorization, then authorization is performed before accounting takes place. If authorization fails for the command, no accounting takes place.

Syntax: `aaa accounting commands <privilege-level> default start-stop radius | tacacs | none`

The `<privilege-level>` parameter can be one of the following:

- **0** – Records commands available at the Super User level (all commands)
- **4** – Records commands available at the Port Configuration level (port-config and read-only commands)
- **5** – Records commands available at the Read Only level (read-only commands)

Configuring RADIUS accounting for system events

You can configure RADIUS accounting to record when system events occur on the ServerIron. System events include rebooting and when changes to the active configuration are made.

The following command causes an Accounting Start packet to be sent to the RADIUS accounting server when a system event occurs, and a Accounting Stop packet to be sent when the system event is completed.

```
ServerIronADX(config)# aaa accounting system default start-stop radius
```

Syntax: `aaa accounting system default start-stop radius | tacacs+ | none`

Configuring an interface as the source for all RADIUS packets

You can designate the lowest-numbered IP address configured on an Ethernet port, [POS port](#), loopback interface, or virtual interface as the source IP address for all RADIUS packets from the Layer 3 Switch. Identifying a single source IP address for RADIUS packets provides the following benefits:

- If your RADIUS server is configured to accept packets only from specific links or IP addresses, you can use this feature to simplify configuration of the RADIUS server by configuring the ServerIron to always send the RADIUS packets from the same link or source address.
- If you specify a loopback interface as the single source for RADIUS packets, RADIUS servers can receive the packets regardless of the states of individual links. Thus, if a link to the RADIUS server becomes unavailable but the client or server can be reached through another link, the client or server still receives the packets, and the packets still have the source IP address of the loopback interface.

The software contains separate CLI commands for specifying the source interface for Telnet, TACACS or TACACS+, and RADIUS packets. You can configure a source interface for one or more of these types of packets.

To specify an Ethernet [or POS](#) port or a loopback or virtual interface as the source for all RADIUS packets from the device, use the following CLI method. The software uses the lowest-numbered IP address configured on the port or interface as the source IP address for RADIUS packets originated by the device.

To specify the lowest-numbered IP address configured on a virtual interface as the device's source for all RADIUS packets, enter commands such as the following.

```
ServerIronADX(config)# interface ve 1
ServerIronADX(config-vif-1)# ip address 10.0.0.3/24
ServerIronADX(config-vif-1)# exit
ServerIronADX(config)# ip radius source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.3/24 to the interface, then designate the interface as the source for all RADIUS packets from the Layer 3 Switch.

Syntax: `ip radius source-interface ethernet <portnum> | pos <portnum> | loopback <num> | ve <num>`

The `<num>` parameter is a loopback interface or virtual interface number. If you specify an Ethernet or POS port, the `<portnum>` is the port's number (including the slot number, if you are configuring a Chassis device).

Displaying RADIUS configuration information

The **show aaa** command displays information about all TACACS or TACACS+ and RADIUS servers identified on the device.

Example

```
ServerIronADX# show aaa
Tacacs+ key: brocade
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server: 207.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

The following table describes the RADIUS information displayed by the **show aaa** command.

TABLE 12 Output of the show aaa command for RADIUS

Field	Description
Radius key	The setting configured with the radius-server key command. At the Super User privilege level, the actual text of the key is displayed. At the other privilege levels, a string of periods (...) is displayed instead of the text.
Radius retries	The setting configured with the radius-server retransmit command.
Radius timeout	The setting configured with the radius-server timeout command.
Radius dead-time	The setting configured with the radius-server dead-time command.

TABLE 12 Output of the show aaa command for RADIUS (Continued)

Field	Description
Radius Server	For each RADIUS server, the IP address, and the following statistics are displayed: <ul style="list-style-type: none"> • Auth PortRADIUS authentication port number (default 1645) • Acct PortRADIUS accounting port number (default 1646) • opensNumber of times the port was opened for communication with the server • closesNumber of times the port was closed normally • timeoutsNumber of times port was closed due to a timeout • errorsNumber of times an error occurred while opening the port • packets inNumber of packets received from the server • packets outNumber of packets sent to the server
connection	The current connection status. This can be “no connection” or “connection active”.

The **show web** command displays the privilege level of Web Management Interface users.

Example

```
ServerIronADX(config)# show web
User                               Privilege   IP address
set                                0           192.168.1.234
```

Syntax: show web

Configuring authentication-method lists

To implement one or more authentication methods for securing access to the device, you configure authentication-method lists that set the order in which the authentication methods are consulted.

In an authentication-method list, you specify the access method (Telnet, Web, SNMP, and so on) and the order in which the device tries one or more of the following authentication methods:

- Local Telnet login password
- Local password for the Super User privilege level
- Local user accounts configured on the device
- Database on a TACACS or TACACS+ server
- Database on a RADIUS server
- No authentication

NOTE

The TACACS or TACACS+, RADIUS, and Telnet login password authentication methods are not supported for SNMP access.

NOTE

To authenticate Telnet access to the CLI, you also must enable the authentication by entering the **enable telnet authentication** command at the global CONFIG level of the CLI. You cannot enable Telnet authentication using the Web Management Interface.

NOTE

You do not need an authentication-method list to secure access based on ACLs or a list of IP addresses. Refer to [“Using ACLs to restrict remote access”](#) on page 93 or [“Restricting remote access to the device to specific IP addresses”](#) on page 95.

In an authentication-method list for a particular access method, you can specify up to seven authentication methods. If the first authentication method is successful, the software grants access and stops the authentication process. If the access is rejected by the first authentication method, the software denies access and stops checking.

However, if an error occurs with an authentication method, the software tries the next method on the list, and so on. For example, if the first authentication method is the RADIUS server, but the link to the server is down, the software will try the next authentication method in the list.

NOTE

If an authentication method is working properly and the password (and user name, if applicable) is not known to that method, this is not an error. The authentication attempt stops, and the user is denied access.

The software will continue this process until either the authentication method is passed or the software reaches the end of the method list. If the Super User level password is not rejected after all the access methods in the list have been tried, access is granted.

Configuration considerations for authentication-method lists

- For CLI access, you must configure authentication-method lists if you want the device to authenticate access using local user accounts or a RADIUS server. Otherwise, the device will authenticate using only the locally based password for the Super User privilege level.
- When no authentication-method list is configured specifically for Web management access, the device performs authentication using the SNMP community strings:
 - For read-only access, you can use the user name “get” and the password “public”. The default read-only community string is “public”.
 - There is no default read-write community string. Thus, by default, you cannot open a read-write management session using the Web Management Interface. You first must configure a read-write community string using the CLI. Then you can log on using “set” as the user name and the read-write community string you configure as the password. Refer to [“Configuring TACACS or TACACS+ security”](#) on page 106.
- If you configure an authentication-method list for Web management access and specify “local” as the primary authentication method, users who attempt to access the device using the Web Management Interface must supply a user name and password configured in one of the local user accounts on the device. The user **cannot** access the device by entering “set” or “get” and the corresponding SNMP community string.
- For devices that can be managed using [IronView](#), the default authentication method (if no authentication-method list is configured for SNMP) is the CLI Super User level password. If no Super User level password is configured, then access through [IronView](#) is not authenticated. To use local user accounts to authenticate access through [IronView](#), configure an authentication-method list for SNMP access and specify “local” as the primary authentication method.

Examples of authentication-method lists

Example

The following example shows how to configure authentication-method lists for the Web Management Interface, [IronView](#) and the Privileged EXEC and CONFIG levels of the CLI. The primary authentication method for each is "local". The device will authenticate access attempts using the locally configured user names and passwords first.

To configure an authentication-method list for the Web Management Interface, enter a command such as the following.

```
ServerIronADX(config)# aaa authentication web-server default local
```

This command configures the device to use the local user accounts to authenticate access to the device through the Web Management Interface. If the device does not have a user account that matches the user name and password entered by the user, the user is not granted access.

To configure an authentication-method list for [IronView](#), enter a command such as the following.

```
ServerIronADX(config)# aaa authentication snmp-server default local
```

This command configures the device to use the local user accounts to authenticate access attempts through [IronView](#).

To configure an authentication-method list for the Privileged EXEC and CONFIG levels of the CLI, enter the following command.

```
ServerIronADX(config)# aaa authentication enable default local
```

This command configures the device to use the local user accounts to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI.

Example

To configure the device to consult a RADIUS server first to authenticate attempts to access the Privileged EXEC and CONFIG levels of the CLI, then consult the local user accounts if the RADIUS server is unavailable, enter the following command.

```
ServerIronADX(config)# aaa authentication enable default radius local
```

Syntax: `[no] aaa authentication snmp-server | web-server | enable | login default <method1> [<method2>] [<method3>] [<method4>] [<method5>] [<method6>] [<method7>]`

The `snmp-server | web-server | enable | login` parameter specifies the type of access this authentication-method list controls. You can configure one authentication-method list for each type of access.

NOTE

TACACS or TACACS+ and RADIUS are supported only with the **enable** and **login** parameters.

The `<method1>` parameter specifies the primary authentication method. The remaining optional `<method>` parameters specify additional methods to try if an error occurs with the primary method. A method can be one of the values listed in the Method Parameter column in the following table.

TABLE 13 Authentication method values

Method parameter	Description
line	Authenticate using the password you configured for Telnet access. The Telnet password is configured using the enable telnet password... command. Refer to “Setting a Telnet password” on page 100.
enable	Authenticate using the password you configured for the Super User privilege level. This password is configured using the enable super-user-password... command. Refer to “Setting passwords for management privilege levels” on page 100.
local	Authenticate using a local user name and password you configured on the device. Local user names and passwords are configured using the username... command. Refer to “Configuring a local user account” on page 105.
tacacs	Authenticate using the database on a TACACS server. You also must identify the server to the device using the tacacs-server command.
tacacs+	Authenticate using the database on a TACACS+ server. You also must identify the server to the device using the tacacs-server command.
radius	Authenticate using the database on a RADIUS server. You also must identify the server to the device using the radius-server command. Refer to “Configuring RADIUS security” on page 122.
none	Do not use any authentication method. The device automatically permits access.

2 Configuring authentication-method lists

Role Based Management

The Role Based Management (RBM) feature allows users to create different administrative domains and enable user-based access privileges on a ServerIron ADX.

Overview

With this feature, a user can view and/or update configurations, such as virtual servers, real servers, and csw policies, without having the capability of viewing or editing configurations associated with another user. This feature also helps to address "virtualization" requirements.

The existing 3-level user privileges have been expanded to 4 levels. Among them, the existing 3 level, 0 for super user, 4 for port config, and 5 for read only, maintain their current forms. A new privilege level (level 1) is added, and access by the users of this level is controlled by the role based policy. The total number of users that can be created on a ServerIron ADX is 1024.

Depending on the configuration, the following roles can be granted to a user of privilege level 1:

- Viewer of global configurations
- Manager of global configurations
- Manager of one or more contexts
- Operator of one or more contexts
- Viewer of one or more contexts

These roles are applicable only to level-1 users.

Super users have all privileges. The manager automatically has operator and viewer privileges for the specific administrative domain and operator has viewer privileges for the domain.

Within a context, a user can be a manager, operator, or viewer of the following items and their child configuration items:

- real-name-or-ip
- server virtual-name-or-ip
- csw-rule
- csw- policy
- healthck
- server port-policy

The contexts are identified by their names. Up to 256 contexts are supported. For backward compatibility, context-oriented configurations not assigned to a context belong to a build-in default context.

- A context cannot be deleted if it is referenced.
- A resource in a context cannot be bound to a resource in a different context. For example, a virtual server in context c1 cannot be bound to a real server in context c2.

- A resource in a context cannot be deleted if the user is in a different context at the time.
- One default context can be configured for a user if the user has privileges for more than one context.

For simplicity of the configuration, the super user can choose to create some role templates and associate the template with a set of privileges (available privileges are the same as the user level configurations). A user can then be associated with one of the templates, in which case, the user is granted privileges in the template. Then user level privileges can be configured to overwrite the privileges in the template:

- If no privilege is granted for an administrative domain at both the template and user levels, the user has no privileges for the domain.
- If a privilege is granted both at the template and user levels, user level privilege takes precedence.
- If a privilege is granted only in at the template, the user inherits the privilege from the template.
- If a privilege is granted only at the user level, the user has the configured privileges.

Similarly, a default context can be associated with a template. If default context is configured at both template and user level, the user level configuration takes precedence. If a user is not associated with any templates or roles, he has no privileges. The super user can also create a default template with a set of privileges, which is assigned to anyone who does not have templates or privileges assigned.

The following commands can only be executed by super users:

- Copy
- Boot
- Reload/Reload-yes
- asm
- Rconsole (level-1 users will have no access to BP)
- Show server debug
- Show users

The following items can only be created, deleted, or configured by the super users:

- Username
- Context
- Role template

Display of the following commands does not include information for contexts not viewable by the current user:

- Show run
- Show startup
- Write term
- Show server real
- Show server virtual
- Show server bind
- Show server traffic
- Show server session

When privileges for a user are changed after the user login, the user's privilege takes effect immediately

Command Line Interface

After user login, the user is automatically associated with the configured context (if there is only one) or the default context (if there are more than one and a default context is configured). If no context is associated with the user, the user must use the "context <name>" command to select a context before the user can edit context-related configurations.

If a user has the privilege to multiple contexts, the user can use the same **context <name>** command to switch between different contexts.

The **show user role** command is provided to display available contexts and corresponding privilege available for the given user. The same information is also available if the **show who** command is issued.

The **show run context <ctxt-name>** command displays configurations of the given context.

The super user can use "context <name>" to create a new context and "no context <name>" to remove it. The built-in default context (shown as "context default") cannot be removed. In configuration file, "context <name>" separates configurations of each context.

The super user can use the "role template <name>" command to create a new role template.

The super user can use "role default" to create a special template. The roles defined in the template are automatically assigned to any level-1 user to whom no templates or privileges have been assigned.

The following commands can be used to associate roles to a template or a user.

- global all manager|viewer|none
- context <ctxt_name> manager|operator|viewer|none
- default-context <ctxt-name>

A user can also inherit privileges from a template by using the following command:

- role template <tpl_name>

Example

```
ServerIronADX(config)# role template t1
ServerIronADX(config-role-template-t1)# global all viewer
ServerIronADX(config-role-template-t1)# context c1 operator
ServerIronADX(config-role-template-t1)# context c2 manager
ServerIronADX(config-role-template-t1)# context c3 viewer
ServerIronADX(config-role-template-t1)# default-context c2

ServerIronADX(config)# username u1 privilege 1 password passw0rd
ServerIronADX(config-role-user-u1)# global all none
ServerIronADX(config-role-user-u1)# context c4 manager
ServerIronADX(config-role-user-u1)# role template t1
ServerIronADX(config-role-user-u1)# default-context c4
```

The role can only be associated with users with privilege level 1. After a privilege level 1 user is created, the super user can assign roles or make changes to the user by re-issuing the **username** command and entering the role configuration mode for that user.

Integrating RBM with RADIUS and TACACS+

You can configure a ServerIron ADX and its corresponding AAA (Radius or TACACS+) server to have RBM access authenticated from the respective AAA server. The following procedures must be used to configure a ServerIron ADX for RBM authentication by a AAA Server :

- Configure the ServerIron ADX for AAA server authentication
- Configure the contexts on the ServerIron ADX
- Specify the context and role information on the AAA server.

Once these procedures are completed access to RBM on the ServerIron ADX follows this sequence:

1. The client logs into the ServerIron ADX with a Username and Password.
2. The ServerIron ADX requests authentication from the AAA (Radius or TACACS+) server.
3. The RADIUS or TACACS+ server authenticates the user for the approved context for the ServerIron ADX.
4. The ServerIron ADX assigns the approved contexts and roles to the user;
5. If the context or template authenticated by the AAA server is not configured on the ServerIron ADX., the ServerIron ADX ignores the context authentication and a log message is generated.

Configuring the ServerIron ADX for AAA authentication

The following must be configured on the ServerIron ADX for AAA authentication.

- The ServerIron ADX must be configured for authentication by a AAA server. Instructions are provided in [Chapter 2, "Secure Access Management"](#). Sample configurations for RADIUS and TACACS+ are provided in ["Configuring a ServerIron ADX for authentication by a RADIUS server"](#) and ["Configuring a ServerIron ADX for authentication by a TACACS+ server"](#). There is nothing in the AAA configurations that is unique to use with RBM.
- A role template must be configured at the global level and the role template must be associated to a user

Configuring a ServerIron ADX for authentication by a RADIUS server

Procedures for configuring a ServerIron ADX for authentication by a RADIUS server are described in detail in ["Configuring RADIUS security"](#) on page 122. The following example is a standard ServerIron ADX configuration for enabling AAA authentication and authorization by a RADIUS server with the following settings:

- Default login authentication is provided by a RADIUS server.
- Exec authorization is configured on the ServerIron ADX to have the user's privilege level authorized by a RADIUS server.
- RADIUS is set as the primary authentication method for securing access to the ServerIron ADX system.
- The server at IP address 10.10.10.10 is identified as the RADIUS server.
- Encryption is enabled through the "key" parameter

```
ServerIronADX(config)# aaa authentication login default radius
ServerIronADX(config)# aaa authorization exec default radius
```



```
ServerIronADX(config)# radius-server host 10.10.10.10 auth-port 1812 acct-port
1813 default
ServerIronADX(config)# radius-server key brocade
```

Configuring a ServerIron ADX for authentication by a TACACS+ server

Procedures for configuring a ServerIron ADX for authentication by a TACACS+ server are described in detail in [“Configuring TACACS or TACACS+ security”](#) on page 106. The following example is a standard ServerIron ADX configuration for enabling AAA authentication and authorization by a TACACS+ server with the following settings:

- Default authentication for “web-server” access is provided locally.
- Exec authorization is configured to have the user’s privilege level authorized by a TACACS+ server.
- TACACS+ is set as the default authentication method for login
- The server at IP address 10.10.10.10 is identified as the TACACS+ server.

```
ServerIronADX(config)# aaa authentication web-server default local
ServerIronADX(config)# aaa authorization exec default tacacs+
ServerIronADX(config)# aaa authentication login default tacacs+
ServerIronADX(config)# tacacs-server host 10.10.10.10 auth-port 49 default key 1
$D?@d=8
```

Configuring a role template at the global level (RADIUS or TACACS+)

The following commands configure a role template named “brcd” on a ServerIron ADX at the global level and associates that template to the user named “user1.”

```
ServerIronADX(config)# role template brcd
ServerIronADX(config-role-template-r1)# context c1 operator
ServerIronADX(config-role-template-r1)# context c2 manager
ServerIronADX(config-role-template-r1)# context c3 viewer
ServerIronADX(config-role-template-r1)# exit
ServerIronADX(config)# username user1 privilege 1 password pass2
ServerIronADX(config-role-user-user1)# role template brcd
```

Configuring the AAA server to authenticate RBM on a ServerIron ADX

The AAA server must be configured to authenticate RBM on the ServerIron ADX for RADIUS or TACACS as shown in the following example for a typical CentOS LINUX operating system.

Configuring a RADIUS server to authenticate RBM

In the file "/usr/local/etc/raddb/users" a configuration such as the following must be added for the RADIUS server to authenticate RBM. In particular, the settings for "SIContextRole" and "SIRoleTemplate =" must reflect the settings in the specified RBM role template that has been configured on the ServerIron ADX.

```
aaaserver1 Cleartext-Password : = "pass"
          foundry-privilege-Level = 1,
          SIContextRole = "c1+operator;c2+viewer;*c3+manager",
          SIRoleTemplate = "brcd"
```

In this RADIUS server configuration, "*" specifies default pair. Where "*" is not present, the first pair will be the default pair to be considered.

Configuring a TACACS+ server to authenticate RBM

In the file "/usr/local/etc/tacacs.conf" a configuration such as the following must be added for the TACACS+ server to authenticate RBM. In particular, the settings for "brocade-context-role =" and "brocade-role-temp1" must reflect the settings in the specified RBM role template.

```
user=brcd1 {
    default server = permit
    #member = admin
    #Global password
    global = cleartext "pass"
    service = exec {
        foundry-privlvl = 1
        brocade-context-role = rad+viewer
        brocade-role-temp1 = brcd
    }
```

Securing SNMP Access

Simple Network Management Protocol (SNMP) is a set of protocols for managing complex networks. SNMP sends messages, called protocol data units (PDUs), to different parts of a network. An SNMP-compliant device, called an agent, stores data about itself in Management Information Bases (MIBs) and SNMP requesters or managers.

Establishing SNMP community strings

SNMP versions 1 and 2c use community strings to restrict SNMP access. The default passwords for SNMP access are the SNMP community strings configured on the device.

- The default read-only community string is “public”. Use this community string for any SNMP Get, GetNext, or GetBulk request.
- By default, you cannot perform any SNMP Set operations since a read-write community string is not configured.

You can configure as many additional read-only and read-write community strings as you need. The number of strings you can configure depends on the memory on the device. There is no practical limit.

If you delete the startup configuration file, the device automatically re-adds the default “public” read-only community string the next time you load the software.

Encryption of SNMP community strings

Encryption is enabled by default. The software automatically encrypts SNMP community strings. Users with read-only access or who do not have access to management functions in the CLI cannot display the strings. For users with read-write access, the strings are encrypted in the CLI but are shown in the clear in the Web Management Interface.

To display the community strings in the CLI, first use the **enable password-display** command and then use the **show snmp server** command. This will display both the read-only and read-write community strings in the clear.

Adding an SNMP community string

By default, the string is encrypted. To add a community string, enter commands such as the following.

```
ServerIronADX(config)# snmp-server community private rw
```

The command adds the read-write SNMP community string “private”.

Syntax: [no] snmp-server community [0] <string> ro | rw [view <viewname>]
[<standard-acl-name> | <standard-acl-id>]

The *<string>* parameter specifies the community string name. The string can be up to 32 characters long.

The **ro** | **rw** parameter specifies whether the string is read-only (ro) or read-write (rw).

The **view** *<viewstring>* parameter is optional. It allows you to associate a view to the members of this community string. Enter up to 32 alphanumeric characters. If no view is specified, access to the full MIB is granted. The view that you want must exist before you can associate it to a community string. Here is an example of how to use the view parameter in the community string command:

```
ServerIronADX(config)# snmp-server community myread ro view sysview
```

The command in this example associates the view "sysview" to the community string named "myread". The community string has read-only access to "sysview". For information on how create views, see the section "Defining SNMP Views" on page 41-8.

The *<standard-acl-name>* | *<standard-acl-id>* parameter is optional. It allows you to specify which ACL is used to filter the incoming SNMP packets. You can enter either the ACL name or its ID. Here are examples:

```
ServerIronADX(config) # snmp-server community myread ro view sysview 2
ServerIronADX(config) # snmp-server community myread ro view sysview myacl
```

The command in the first example specifies that ACL group 2 filters incoming SNMP packets, whereas the command in the second example uses the ACL group called "myacl" to filter incoming packets.

Displaying the SNMP community strings

To display the community strings in the CLI, first use the **enable-password-display** command and then use the **show snmp server** command. This will display both the read-only and read-write community strings in the clear.

To display the configured community strings, enter the following command at any CLI level:

```
ServerIronADX(config)# show snmp server
Contact:
    Location:
Community(ro): .....
Traps
    Cold start: Enable
    Link up: Enable
    Link down: Enable
    Authentication: Enable
    Locked address violation: Enable
    Power supply state change: Enable
    Fan failure: Enable
    Redundant module state change: Enable
    Temperature warning: Enable
    STP new root: Enable
    STP topology change: Enable
    L4 maximum session limit: Enable
    L4 TCP SYN limit: Enable
    L4 server max connection: Enable
    L4 server up: Enable
    L4 server down: Enable
```

NOTE

If display of the strings is encrypted, the strings are not displayed. Encryption is enabled by default.

Using the user-based security mode

SNMP version 3 (RFC 2570 through 2575) introduces a User-Based Security model (RFC 2574) for authentication and privacy services.

SNMP version 1 and version 2 use community strings to authenticate SNMP access to management modules. This method can still be used for authentication. In SNMP version 3, the User-Based Security model of SNMP can be used to secure against the following threats:

- Modification of information
- Masquerading the identity of an authorized entity
- Message stream modification
- Disclosure of information

Furthermore, SNMP version 3 supports View-Based Access Control Mechanism (RFC 2575) to control access at the PDU level. It defines mechanisms for determining whether or not access to a managed object in a local MIB by a remote principal should be allowed.

Configuring your NMS

To be able to use the SNMP version 3 features, perform the following steps.

1. Make sure that your Network Manager System (NMS) supports SNMP version 3.
2. Configure your NMS agent with the necessary users.
3. Configure the SNMP version 3 features in the ServerIron ADX.

Configuring SNMP version 3 on the ServerIron ADX

To configure SNMP version 3 on the ServerIron ADX, do the following:

1. Enter an engine ID for the management module using the **snmp-server engineid** command if you will not use the default engine ID. See [“Defining the engine ID”](#) on page 150.
2. Create views that will be assigned to SNMP user groups using the **snmp-server view** command.
3. Create ACL groups that will be assigned to SNMP user groups using the **access-list** command. Refer to the “Access Control List” chapter in the *ServerIron ADX Security Guide* for details.
4. Create user groups using the **snmp-server group** command. See [“Defining an SNMP group”](#) on page 150.
5. Create user accounts and associate these accounts to user groups using the **snmp-server user** command. See [“Defining an SNMP user account”](#) on page 151.

If SNMP version 3 is not configured, then community strings by default are used to authenticate access.

Even if SNMP version 3 users are configured on the device, the system will still accept SNMP version 1, 2c and 3 PDUs from the remote manager.

Defining the engine ID

A default engine ID is generated during system start up. The format of the default engine ID is derived from RFC 2571 (Architecture for SNMP frameworks) within the MIB description for object SnmpEngineID.

To determine what the default engine ID of the device is, enter the **show snmp engineid** command and find the following line.

```
Local SNMP Engine ID: 800007c70300e05290ab60
```

See the section ["Displaying the engine ID"](#) on page 153 for details.

The default engine ID guarantees the uniqueness of the engine ID for SNMP version 3. If you want to change the default engine ID, enter a command such as the following:

```
ServerIronADX(config)# snmp-server engineid local 800007c70300e05290ab60
```

Syntax: **[no] snmp-server engineid local** *<hex-string>*

The **local** parameter indicates that engine ID to be entered is the ID of this device, representing an SNMP management entity.

NOTE

Since the current implementation of SNMP version 3 does not support Notification, remote engine IDs cannot be configured at this time.

The *<hex-string>* variable consists of 11 octets, entered as hexadecimal values. Each octet has two hexadecimal characters. The engine ID should contain an even number of hexadecimal characters.

The default engine ID has a maximum of 11 octets:

- Octets 1 through 4 represent the agent's SNMP management private enterprise number as assigned by the Internet Assigned Numbers Authority (IANA). The most significant bit of Octet 1 is "1". For example, "000007c7" is the ID for Brocade in hexadecimal. With Octet 1 always equal to "1", the first four octets in the default engine ID is always "800007c7" (which is 1991 in decimal).
- Octet 5 is always 03 in hexadecimal and indicates that the next set of values represent a MAC address.
- Octets 6 through 11 form the MAC address of the lowest port in the management module.

NOTE

Engine ID must be a unique number among the various SNMP engines in the management domain. Using the default engine ID ensures the uniqueness of the numbers.

Defining an SNMP group

SNMP groups map SNMP users to SNMP views. For each SNMP group, you can configure a read view, a write view, or both. Users who are mapped to a group will use its views for access control.

To configure an SNMP user group, enter a command such as the following:

```
ServerIronADX(config)# snmp-server group admin v3 auth read all write all
```

Syntax: **[no] snmp-server group** *<groupname>*
v1 | v2c | v3

```
auth | noauth | priv
[access <standard-acl-id>] [read <viewstring>] [write <viewstring>] [notify <viewname>]
```

NOTE

This command is not used for SNMP version 1 and SNMP version 2. In these versions, groups and group views are created internally using community strings. (See [“Establishing SNMP community strings”](#) on page 147.) When a community string is created, two groups are created, based on the community string name. One group is for SNMP version 1 packets, while the other is for SNMP version 2 packets.

The **group** *<groupname>* parameter defines the name of the SNMP group to be created.

The **v1**, **v2c**, or **v3** parameter indicates which version of SNMP is used. In most cases, you will be using v3, since groups are automatically created in SNMP versions 1 and 2 from community strings.

The **auth** | **noauth** parameter determines whether authentication is required for accessing the supported views. If **auth** is selected, then only authenticated packets are allowed to access the view specified for the user group. Selecting **noauth** means that no authentication is required to access the specified view. Selecting **priv** means that an authentication password is required from the users.

The **auth** | **noauth** | **priv** parameter is available when you select v3, not v1 or v2.

The **access** *<standard-acl-id>* parameter is optional. It allows incoming SNMP packets to be filtered based on the standard ACL attached to the group.

The **read** *<viewstring>* | **write** *<viewstring>* parameter is optional. It indicates that users who belong to this group have either read or write access to the MIB.

The **notify** *<viewname>* parameter is optional. It allows trap notifications to be encrypted and sent to target hosts.

The *<viewstring>* variable is the name of the view to which the SNMP group members have access. If no view is specified, then the group has no access to the MIB.

The value of *<viewstring>* is defined by using the **snmp-server view** command. The SNMP agent comes with the "all" view, the default view that provides access to the entire MIB; however, it must be specified when creating the group. The "all" view also lets SNMP version 3 be backwards compatible with SNMP version 1 and version 2.

NOTE

If you plan to use a view other than the "all" view, that view must have been configured before you create the user group. See [“Defining SNMP Views”](#) on page 41-8, for details on the **include** | **exclude** parameters.

Defining an SNMP user account

The **snmp-server user** command does the following.

- Creates an SNMP user.
- Defines the group to which the user will be associated.
- Defines the type of authentication to be used for SNMP access by this user.

Here is an example of how to create the account:

4 Using the user-based security mode

```
ServerIronADX(config)# snmp-s user bob admin v3 access 2 auth md5 bobmd5 priv des bobdes
```

The CLI for creating SNMP version 3 users has been updated as follows.

Syntax: `[no] snmp-server user <name> <groupname> v3`
`[[access <standard-acl-id>]`
`[[encrypted] auth md5 <md5-password> | sha <sha-password>`
`[priv [encrypted] des <des-password-key> | aes <aes-password-key>]]]`

The *<name>* parameter defines the SNMP user name or security name used to access the management module.

The *<groupname>* parameter identifies the SNMP group to which this user is associated or mapped. All users must be mapped to an SNMP group. Groups are defined using the **snmp-server group** command.

NOTE

The SNMP group to which the user account will be mapped should be configured before creating the user accounts; otherwise, the group will be created without any views. Also, ACL groups must be configured before configuring user accounts.

The **v3** parameter is required.

The **access** *<standard-acl-id>* parameter is optional. It indicates that incoming SNMP packets are filtered based on the ACL attached to the user account.

NOTE

The ACL specified in a user account overrides the ACL assigned to the group to which the user is mapped. If no ACL is entered for the user account, the ACL configured for the group is used to filter packets.

The encrypted parameter means that the MD5 or SHA password will be a digest value. MD5 has 16 octets in the digest. SHA has 20. The digest string has to be entered as a hexadecimal string. In this case, the agent need not generate any explicit digest. If the encrypted parameter is not used, the user is expected to enter the authentication password string for MD5 or SHA. The agent converts the password string to a digest, as described in RFC 3414.

The optional **auth md5 | sha** parameter defines the type of encryption the user must have to be authenticated. The choices are MD5 and SHA encryption (the two authentication protocols used in SNMP version 3).

The *<md5-password>* and *<sha-password>* define the password the user must use to be authenticated. These password must have a minimum of 8 characters. If the encrypted parameter is used, then the digest has 16 octets for MD5 or 20 octets for SHA.

NOTE

Once a password string is entered, the generated configuration displays the digest (for security reasons), not the actual password.

The **priv [encrypted]** parameter is optional after you enter the md5 or sha password. The **priv** parameter specifies the encryption that is used to encrypt the privacy password. If the **encrypted** keyword is used, do the following:

- If DES is the privacy protocol to be used, enter **des** *<des-password-key>* and enter a 16-octet DES key in hexadecimal format for the *<des-password-key>*. If you include the **encrypted** keyword, enter a password string of at least 8 characters.

- If AES is the privacy protocol to be used, enter **aes** and an *<aes-password-key>*. Enter either 12 (for a small key) or 16 (for a big key) characters for the *<aes-password-key>*. If you include the **encrypted** keyword, enter a password string containing 32 hexadecimal characters.

Displaying the engine ID

To display the engine ID of a management module, enter a command such as the following:

```
ServerIronADX(config)# show snmp engineid
Local SNMP Engine ID: 800007c70300e05290ab60
Engine Boots: 3
Engine time: 5
```

Syntax: `show snmp engineid`

The engine ID identifies the source or destination of the packet.

The engine boots represents the number of times that the SNMP engine reinitialized itself with the same engine ID. If the engineID is modified, the boot count is reset to 0.

The engine time represents the current time with the SNMP agent.

Displaying SNMP groups

To display the definition of an SNMP group, enter a command such as the following:

```
ServerIronADX(config)# show snmp group
groupname = exceptifgrp
security model = v3
security level = authNoPriv
ACL id = 2
readview = exceptif
writeview = <none>
```

Syntax: `show snmp group`

The value for security level can be one of the following.

Security level	Authentication
<none>	If the security model shows v1 or v2, then security level is blank. User names are not used to authenticate users; community strings are used instead.
noauthNoPriv	Displays if the security model shows v3 and user authentication is by user name only.
authNoPriv	Displays if the security model shows v3 and user authentication is by user name and the MD5 or SHA algorithm.
authPriv	Authentication uses MD5 or SHA. Encryption uses DES and AES protocol.

Displaying user information

To display the definition of an SNMP user account, enter a command such as the following.

```
ServerIronADX(config)# show snmp user
username = bob
acl id = 0
group = bobgroup
security model = v3
group acl id = 0
authtype = md5
authkey = ad172674ebc09cd9448c8276db0d12f8
privtype = aes
privkey = 3c154b47996534b22b22758e23f9a71a
engine ID= 800007c703000cdbf48a00
```

Syntax: show snmp user

Interpreting varbinds in report packets

If an SNMP version 3 request packet is to be rejected by an SNMP agent, the agent sends a report packet that contains one or more varbinds. The varbinds contain additional information, showing the cause of failures. An SNMP manager application decodes the description from the varbind. The following table presents a list of varbinds supported by the SNMP agent.

Varbind object identifier	Description
1. 3. 6. 1. 6. 3. 11. 2. 1. 3. 0	Unknown packet data unit.
1. 3. 6. 1. 6. 3. 12. 1. 5. 0	The value of the varbind shows the engine ID that needs to be used in the snmp-server engineid command
1. 3. 6. 1. 6. 3. 15. 1. 1. 1. 0	Unsupported security level.
1. 3. 6. 1. 6. 3. 15. 1. 1. 2. 0	Not in time packet.
1. 3. 6. 1. 6. 3. 15. 1. 1. 3. 0	Unknown user name. This varbind can also be generated if either the: <ul style="list-style-type: none">Configured ACL for the user filters out the packet.Group associated with the user is unknown.
1. 3. 6. 1. 6. 3. 15. 1. 1. 4. 0	Unknown engine ID. The value of this varbind would be the correct authoritative engineID that should be used.
1. 3. 6. 1. 6. 3. 15. 1. 1. 5. 0	Wrong digest.
1. 3. 6. 1. 6. 3. 15. 1. 1. 6. 0	Decryption error.

SNMP v3 configuration examples

The examples below shows how to configure SNMP v3.

Simple SNMP v3 configuration

```

ServerIronADX(config)# snmp-s group admingrp v3 priv read all write all notify all
ServerIronADX(config)# snmp-s user adminuser admingrp v3 auth md5 <auth password>
priv <privacy password>
ServerIronADX(config)# snmp-s host <dest-ip> adminuser

```

More detailed SNMP v3 configuration

```

ServerIronADX(config)# snmp-server view system system included
ServerIronADX(config)# snmp-server community ..... ro
ServerIronADX(config)# snmp-server community ..... rw
ServerIronADX(config)# snmp-server contact isc-operations
ServerIronADX(config)# snmp-server location sdh-pillbox
ServerIronADX(config)# snmp-server host 128.91.255.32 .....
ServerIronADX(config)# snmp-server group ops v3 priv read internet write system
ServerIronADX(config)# snmp-server group admin v3 priv read internet write
internet
ServerIronADX(config)# snmp-server group restricted v3 priv read internet
ServerIronADX(config)# snmp-server user ops ops v3 encrypted auth md5
ab8e9cd6d46e7a270b8c9549d92a069 priv encrypted des
0e1b153303b6188089411447dbc32de
ServerIronADX(config)# snmp-server user admin admin v3 encrypted auth md5
0d8a2123f91bfbd8695fef16a6f4207b priv encrypted des
18e0cf359fce4fcd60df19c2b6515448
ServerIronADX(config)# snmp-server user restricted restricted v3 encrypted auth
md5 261fd8f56a3ad51c8bcecle4609f54dc priv encrypted des
d32e66152f89de9b2e0cb17a65595f43

```

4 SNMP v3 configuration examples

Capacity on Demand

You can upgrade an existing ServerIron ADX to increase system capacity by purchasing and applying a new software license. This chapter describes the upgrades available the process required to implement those upgrade.

Software license terminology

This section defines the key terms used in this chapter.

- **Entitlement certificate** – The proof-of-purchase certificate (*paper-pack*) issued by Brocade when a license is purchased. The certificate contains a unique *transaction key* that is used in conjunction with the *License ID* of the Brocade device to generate and download a software license from the Brocade software portal.
- **License file** – The file produced by the Brocade software portal when the license is generated. The file is uploaded to the Brocade device and controls access to a *licensed feature* or feature set.
- **License ID (LID)** – This is a number that uniquely identifies the Brocade device. The LID is used in conjunction with a *transaction key* to generate and download a software license from the Brocade software portal. The software license is tied to the LID of the Brocade device for which the license was ordered and generated.
- **Licensed feature** – Any hardware or software feature or set of features that require a valid software license in order to operate on the device.
- **Transaction key** – This unique key, along with the *LID*, is used to generate a software license from the Brocade software portal. The transaction key is issued by Brocade when a license is purchased. The transaction key is delivered according to the method specified when the order is placed:
 - **Paper-pack** – The transaction key is recorded on an *entitlement certificate*, which is mailed to the customer.
 - **Electronic** – The transaction key is contained in an e-mail, which is instantly sent to the customer after the order is placed. The customer will receive the e-mail generally within a few minutes after the order is placed, though the timing will vary depending on the network, internet connection, etc.

If a delivery method was not specified at the time of the order, the key will be delivered via paper-pack.

Software-based licensing overview

Software and hardware features of both fixed-configuration (ServerIron ADX 1000 series) and chassis (ServerIron ADX 4000 and 10000) ServerIron ADX application switches can be obtained at time of purchase or upgraded later through software-based licensing. At time of purchase a ServerIron ADX application switch is configured with a base license pre-installed at the factory as determined by the SKU ordered (see [Table 14](#) and [Table 16](#) for reference). You can upgrade an existing ServerIron ADX application switch at a later date to provide more functionality as described in the following.

ServerIron ADX 1000 series – The following hardware and software features are enabled depending on the licence that your application switch has configured.

Application Processor – The ServerIron ADX 1000 series application switch is equipped with 4 application processors. The minimum configuration enables 1 application processor and base licences are available that enable 1, 2 or 4 application processors. Systems with 1 or 2 APs can be upgraded to enable 2 or 4.

1 Gbps copper ports – The ServerIron ADX 1000 series application switch is equipped with sixteen 1 Gbps copper ports. The minimum configuration enables 8 of these ports and base licences are available that enable 16. A system with an 8 port licence can be upgraded to support 16 ports.

Hardware-based SSL offload – The ServerIron ADX 1000 series application switch is equipped with an embedded SSL hardware module. This module is not enabled in the minimum configuration. Base licences are available that enable this module or you can upgrade your system to enable it.

10 Gbps ports – The ServerIron ADX 1000 series application switch is equipped with two 10 Gbps ports. Base licences are available with 0 or 2 ports enabled. You can upgrade a system to enable 2 ports.

Premium software – Base licences and upgrades are available that enable Layer-3 and Global Load Balancing features for the ServerIron ADX router image.

ServerIron ADX 4000 and 10000 series – Hardware features are enabled for chassis devices for any module installed on the system. The following software features are enabled depending on the licence that your application switch has configured.

Premium software – Base licences and upgrades are available that enable Layer-3 and Global Load Balancing features for the ServerIron ADX router image.

ServerIron ADX 4000 ASM4 Bundle – A special bundle is available with the ServerIron ADX 4000 chassis that includes the following features.

Application Processor – The ServerIron ADX ASM4 Bundle is configured with an Application Switch module with 4 of the available 8 Application Processors enabled. This module is designated as an ASM4. It will only work in a ServerIron ADX 4000 chassis and only in a single ASM configuration. If an additional ASM4 module is installed, only the ASM4 module installed in the ASM 1 slot will be accepted. If an ASM8 module is installed in a system with an ASM4 module, the ASM4 will be rejected and the ASM8 module will be accepted by the system. You can purchase an upgrade license to increase the capability of an ASM4 to that of an ASM8. The ASM4 license is tied to the ASM module and not to the chassis. Consequently, a license can be purchased to upgrade the ASM4 to use all eight available Application Processors with all functionality and performance options enabled. With this license installed, the ASM module functions and is recognized as an ASM8 module.

NOTE

The license that upgrades the ASM4 to an ASM8 is not available as a trial license.

12 x 1 GF Interface Module – A 12 x 1 GF Interface Module equipped with 8 Copper SFPs is included in the ServerIron ADX ASM4 Bundle.

Management Module – A single Management module is included in the ServerIron ADX ASM4 Bundle.

Power Supplies – Two AC Power Supplies are included in the ServerIron ADX ASM4 Bundle.

Premium software – The Premium software license that enables Layer-3 and Global Load Balancing features for the ServerIron ADX router image is included in the ServerIron ADX ASM4 Bundle.

NOTE

Upgrade options are described in [Table 15](#) and [Table 17](#).

When a license is ordered separately (not pre-installed), an *entitlement certificate* or e-mail, along with a *transaction key*, are issued to the customer by Brocade as proof of purchase. The *transaction key* and *LID* of the Brocade device are used to generate a license key from the Brocade software licensing portal. The license key is contained within a *license file*, which is downloaded to the customer's PC, where the file can then be transferred to a TFTP or SCP server, then uploaded to the Brocade device.

Once a license is installed on the Brocade device a reload is required for the new functions to be enabled on the switch.

License types

The following license types are supported on ServerIron ADX devices:

- **Trial license** – this enables a license-controlled feature to run on the device on a temporary basis. A trial license enables demonstration and evaluation of a licensed feature and can be valid for a period of 45 days. For more information about a trial license, see [“Using a trial license”](#) on page 170.
- **Unlimited license** – this enables a license-controlled feature to run on the device indefinitely.

Licensed features and SKU numbers

[Table](#) lists the base configurations available for ServerIron ADX 1000 series application switches by SKU number.

TABLE 14 ServerIron ADX 1000 series base configurations

Base SKUs	Enabled features
SI-1008-1	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 1 Application Processor (AP) • eight 1 Gbps ports • Supports switch image only

TABLE 14 ServerIron ADX 1000 series base configurations

Base SKUs	Enabled features
SI-1008-1-DC	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 1 Application Processor (AP) • eight 1 Gbps ports • Supports switch image only • DC power
SI-1008-1-PREM	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 1 Application Processor (AP) • eight 1 Gbps ports • Supports PREM (L3) license • Supports switch and router images
SI-1008-1-SSL	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 1 Application Processor (AP) • eight 1 Gbps ports • Supports switch image only • Supports SSL hardware acceleration
SI-1008-1-SSL-PREM	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 1 Application Processor (AP) • eight 1 Gbps ports • Supports switch and router images • Supports SSL hardware acceleration • Supports PREM (L3) license
SI-1016-2	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 2 Application Processors (AP) • sixteen 1 Gbps ports • Supports switch image only
SI-1016-2-DC	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 2 Application Processors (AP) • sixteen 1 Gbps ports • Supports switch image only • DC power
SI-1016-2-PREM	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 2 Application Processors (AP) • Sixteen 1 Gbps ports • Supports PREM (L3 + GSLB) license • Supports switch and router images
SI-1016-2-SSL	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 2 Application Processors (AP) • Sixteen 1 Gbps ports • Supports switch image only • Supports SSL hardware acceleration
SI-1016-2-SSL-PREM	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 2 Application Processors (AP) • Sixteen 1 Gbps ports • Supports PREM (L3 + GSLB) license • Supports switch and router images • Supports SSL hardware acceleration

TABLE 14 ServerIron ADX 1000 series base configurations

Base SKUs	Enabled features
SI-1016-4	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 4 Application Processors (AP) • Sixteen 1 Gbps ports • Supports switch image only
SI-1016-4-PREM	<ul style="list-style-type: none"> • 4 Application Processors (AP) • Sixteen 1 Gbps ports • Supports PREM (L3 + GSLB) license • Supports switch and router images
SI-1016-4-SSL	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 4Application Processors (AP) • Sixteen 1 Gbps ports • Supports switch image only • Supports SSL hardware acceleration
SI-1016-4-SSL-PREM	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 4 Application Processors (AP) • Sixteen 1 Gbps ports • Supports PREM (L3 + GSLB) license • Supports switch and router images • Supports SSL hardware acceleration
SI-1216-4	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 4 Application Processors (AP) • Sixteen 1 Gbps ports • Supports switch image only • two 10 Gbps ports
SI-1216-4-PREM	<ul style="list-style-type: none"> • 4 Application Processors (AP) • Sixteen 1 Gbps ports • Supports PREM (L3 + GSLB) license • Supports switch and router images • two 10 Gbps ports
SI-1216-4-SSL	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 4Application Processors (AP) • Sixteen 1 Gbps ports • Supports switch image only • Supports SSL hardware acceleration • two 10 Gbps ports
SI-1216-4-SSL-PREM	ServerIron ADX 1000 switch: <ul style="list-style-type: none"> • 4 Application Processors (AP) • Sixteen 1 Gbps ports • Supports PREM (L3 + GSLB) license • Supports switch and router images • Supports SSL hardware acceleration • two 10 Gbps ports

Table 15 lists the upgrade options available by license for ServerIron ADX 1000 series application switches.

TABLE 15 Upgrade licenses available for ServerIron ADX 1000 series application switches

5 Licensed features and SKU numbers

Licence	Eligible Base SKU	Equivalent Licence after upgrade
ADX-1008-1-LIC-2PPLS (Perpetual License)	SI-1008-1	SI-1016-2
	SI-1008-1-DC	SI-1016-2
ADX-1008-1-TRP-2PPLS (Trial License)	SI-1008-1-PREM	SI-1016-2-PREM
	SI-1008-1-SSL	SI-1016-2-SSL
	SI-1008-1-SSL-PREM	SI-1016-2-SSL-PREM
ADX-1K-LIC-PREM (Perpetual License)	SI-1008-1	SI-1008-1-PREM
	SI-1008-1-DC	SI-1008-1-PREM
ADX-1K-TRL-PREM (Trial License)	SI-1008-1-SSL	SI-1008-1-SSL-PREM
	SI-1016-2	SI-1016-2-PREM
	SI-1016-2-DC	SI-1016-2-PREM
	SI-1016-2-SSL	SI-1016-2-SSL-PREM
	SI-1016-4	SI-1016-4-PREM
	SI-1016-4-SSL	SI-1016-4-SSL-PREM
	SI-1216-4	SI-1216-4-PREM
	SI-1216-4-SSL	SI-1216-4-SSL-PREM
ADX-1-2-1K-LIC-SSL (Perpetual License)	SI-1008-1	SI-1008-1-SSL
	SI-1008-1-DC	SI-1008-1-SSL
ADX-1-2-1K-TRL-SSL (Trial License)	SI-1008-1-PREM	SI-1008-1-SSL-PREM
	SI-1016-2	SI-1016-2-SSL
	SI-1016-2-DC	SI-1016-2-SSL
	SI-1016-2-PREM	SI-1016-2-SSL-PREM
ADX-4-1K-LIC-SSL (Perpetual License)	SI-1016-4	SI-1016-4-SSL
	SI-1016-4-PREM	SI-1016-4-SSL-PREM
ADX-4-1K-TRL-SSL (Trial License)	SI-1216-4	SI-1216-4-SSL
	SI-1216-4-PREM	SI-1216-4-SSL-PREM
ADX-1016-2-LIC-4P (Perpetual License)	SI-1016-2	SI-1016-4
	SI-1016-2-DC	SI-1016-4
ADX-1016-2-TRL-4P (Trial License)	SI-1016-2-PREM	SI-1016-4-PREM
	SI-1016-2-SSL	SI-1016-4-SSL
	SI-1016-2-SSL-PREM	SI-1016-4-SSL-PREM
ADX-1016-2-LIC-4P10G (Perpetual License)	SI-1016-2	SI-1216-4
	SI-1016-2-DC	SI-1216-4
ADX-1016-2-TRL-4P10G (Trial License)	SI-1016-2-PREM	SI-1216-4-PREM
	SI-1016-2-SSL	SI-1216-4-SSL
	SI-1016-2-SSL-PREM	SI-1216-4-SSL-PREM

Licence	Eligible Base SKU	Equivalent Licence after upgrade
ADX-1016-4-LIC-10G (Perpetual License)	SI-1016-4	SI-1216-4
	SI-1016-4-PREM	SI-1216-4-PREM
ADX-1016-4-TRL-10G (Trial License)	SI-1016-4-SSL	SI-1216-4-SSL
	SI-1016-4-SSL-PREM	SI-1216-4-SSL-PREM

Table 16 lists the base configurations available for ServerIron ADX chassis-based application switches by SKU number.

TABLE 16 ServerIron ADX chassis-based systems base configurations

Base SKU	Enabled features
SI-4000	ServerIron ADX 4000 chassis <ul style="list-style-type: none"> Supports switch image only
SI-4000-DC	ServerIron ADX 4000 chassis: <ul style="list-style-type: none"> DC power Supports switch image only
SI-4000-PREM	ServerIron ADX 4000 chassis: <ul style="list-style-type: none"> Supports PREM (L3 + GSLB) license Supports switch and router images
SI-4000-ASM4-P12-B	ServerIron ADX 4000 chassis: <ul style="list-style-type: none"> one ASM4 Application Switch Module one Management module one 12 x 1 GF Interface Module equipped with 8 Copper SFPs 2 AC power supplies Supports PREM (L3 + GSLB) license Supports switch and router images
SI-8000	ServerIron ADX 8000 chassis <ul style="list-style-type: none"> Supports switch image only
SI-8000-DC	ServerIron ADX 8000 chassis: <ul style="list-style-type: none"> DC power Supports switch image only
SI-8000-PREM	ServerIron ADX 8000 chassis: <ul style="list-style-type: none"> Supports PREM (L3 + GSLB) license Supports switch and router images
SI-10000	ServerIron ADX 10000 chassis <ul style="list-style-type: none"> Supports switch image only
SI-10000-DC	ServerIron ADX 10000 chassis: <ul style="list-style-type: none"> DC power Supports switch image only
SI-10000-PREM	ServerIron ADX 10000 chassis: <ul style="list-style-type: none"> Supports PREM (L3 + GSLB) license Supports switch and router images

Table 17 lists the upgrade options available by license for ServerIron ADX chassis-based application switches

TABLE 17 Upgrade licenses available for ServerIron ADX chassis-based application switches

Licence	Eligible Base SKU	Equivalent Licence after upgrade
ADX-CH-LIC-PREM (Perpetual License)	SI-4000	SI-4000-PREM
	SI-4000-DC	SI-4000-PREM
	SI-8000	SI-8000-PREM
	SI-8000-DC	SI-8000-PREM
ADX-CH -TRL-PREM (Trial License)	SI-10000	SI-10000-PREM
	SI-10000-DC	SI-10000-PREM
ADX-CH-LIC-ASM4-8 (Perpetual License)	ASM4	ASM8
There is no trial license available for this upgrade.	Please note, the ASM4 and ASM8 licenses are tied to the module and not the chassis.	

Licensing rules

This section lists the software licensing rules and caveats related to the Brocade devices that support software-based licensing.

General notes

The following licensing rules apply to all ServerIron ADX devices that support software licensing:

- A license is tied to the unique LID of the management module or fixed configuration switch for which the license was ordered. Therefore, a license can be used on one particular device only. It cannot be used on any other device. (not applicable to the ASM4)
- More than one license can be installed per device. (not applicable to the ServerIron ADX chassis devices)
- More than one trial license can be in effect at the same time, as long as each trial license applies to a unique licensed feature. (not applicable to the ServerIron ADX chassis devices)

Configuration tasks

This section describes the configuration tasks for generating and obtaining a software license, then installing it on the Brocade device. Perform the tasks in the order listed in [Table 18](#).

TABLE 18 Configuration tasks for software licensing

Configuration task	See...
1 Order the desired license.	For a list of available licenses and associated part numbers, see "Licensed features and SKU numbers" on page 159.
2 When you receive the transaction key, retrieve the LID of the Brocade device. If you received the transaction key by way of paper-pack, record the LID on the entitlement certificate in the space provided for your records.	"Viewing the License ID (LID)" on page 174
3 Log in to the Brocade software portal to generate and obtain the license file.	"Obtaining a license" on page 165
4 Upload the license file to the Brocade device.	"Installing a license file" on page 169
5 Verify that the license is installed.	"Verifying the license file installation" on page 170
6 For Premium upgrades, upload the Layer-3 routing software image to the device.	Refer to the release notes

Obtaining a license

The procedures in this section show how to generate and obtain a software license.

1. Order a license for the desired licensed feature. Refer to [Table 15](#) and [Table 17](#) for a list of valid part numbers and licensed features.

NOTE

To order and obtain a *trial license*, contact Brocade Technical Support.

2. When you receive the paper-pack or electronic transaction key, retrieve the LID of your Brocade device by entering the **show version** command on the device. Example command output is shown in ["Viewing the License ID \(LID\)"](#) on page 174."

If you received a paper-pack transaction key, write the LID in the space provided on the entitlement certificate for your records.

NOTE

Do not discard the entitlement certificate or e-mail with electronic key. Keep it in a safe place in case it is needed for technical support or product replacement (RMAs).

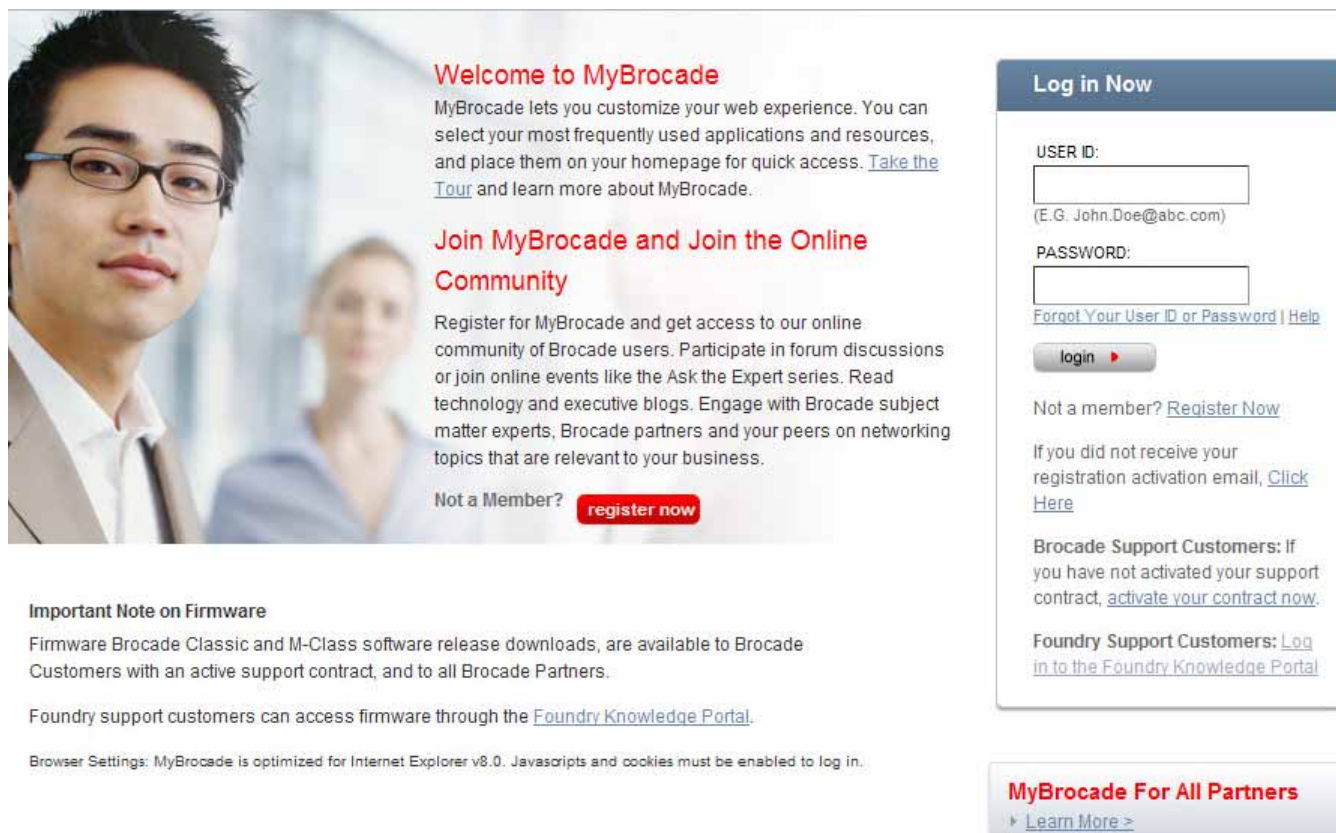
3. Log in to the Brocade software portal at <http://swportal.brocade.com/wps/myportal> and complete the software license request.

NOTE

If you do not have a login ID and password, complete the self registration process before proceeding to log in.

Figure 5 shows the **Software Portal Login** window.

FIGURE 5 Brocade Software Portal Login window



Welcome to MyBrocade

MyBrocade lets you customize your web experience. You can select your most frequently used applications and resources, and place them on your homepage for quick access. [Take the Tour](#) and learn more about MyBrocade.

Join MyBrocade and Join the Online Community

Register for MyBrocade and get access to our online community of Brocade users. Participate in forum discussions or join online events like the Ask the Expert series. Read technology and executive blogs. Engage with Brocade subject matter experts, Brocade partners and your peers on networking topics that are relevant to your business.

Not a Member? [register now](#)

Log in Now

USER ID:

 (E.G. John.Doe@abc.com)

PASSWORD:

[Forgot Your User ID or Password](#) | [Help](#)

[login](#)

Not a member? [Register Now](#)

If you did not receive your registration activation email, [Click Here](#)

Brocade Support Customers: If you have not activated your support contract, [activate your contract now](#).

Foundry Support Customers: [Log in to the Foundry Knowledge Portal](#)

MyBrocade For All Partners

[Learn More >](#)

Important Note on Firmware

Firmware Brocade Classic and M-Class software release downloads, are available to Brocade Customers with an active support contract, and to all Brocade Partners.

Foundry support customers can access firmware through the [Foundry Knowledge Portal](#).

Browser Settings: MyBrocade is optimized for Internet Explorer v8.0. Javascripts and cookies must be enabled to log in.

From the **License Management** menu, select **Brocade IP/ADP > License Generation with Transaction key**. The IP/ADP License Generation dialog box displays.

FIGURE 6 License Management Welcome window

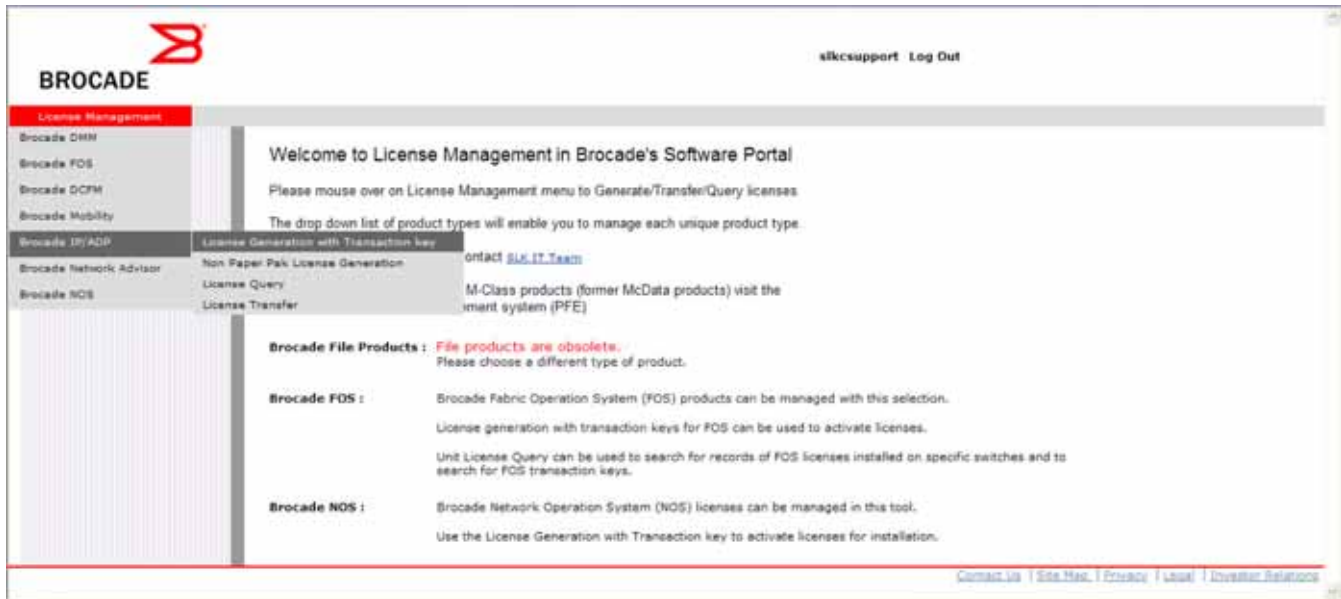


Figure 7 shows the **IP/ADP License Generation** dialog box for generating a license using a transaction key and LID.

FIGURE 7 IP/ADP License Generation window

BROCADE support1 Log Out

License Management

IP/ADP Licence Generation
Please check the Unique ID to make sure it is correct! License install failure may result!

Customer Information

* Indicates required field

Customer email ID*

Site Name

Technical Contact

Company Name

City

State/Province

Zip/Postal Code

Country

Phone

E-mail Notification Information

Self

☐ Other e-mail address(es)

Unit Information

Unique ID Type*

Unit's Unique ID*

Transaction Key*

☐ I have read and accept the [Brocade End User License Agreement](#)

[Contact Us](#)

Enter the required information.

- For a description of the field, move the pointer over the field.
- An asterisk next to a field indicates that the information is required.

NOTE

You can generate multiple licenses at a time. For each license request, enter the **Transaction Key** and the **Unit's Unique ID**, and click **Add**.

When you have finished entering the required information, read the Brocade End User License Agreement, and select the **I have read and accept** check box.

Click the **Generate** button to generate the license. Figure 8 shows the results window, which displays an order summary and the results of the license request.

- If the license request was successful, the **Status** field shows “Success” and the **License File** field contains a hyperlink to the generated license file. The license file is automatically sent by e-mail to the specified customer e-mail address.
- If the license request fails, the **Status** field indicates the reason it failed and the action to be taken.

FIGURE 8 IP/ADP License Generation Results window

License Management

IP/ADP Licence Generation - Result

Customer Information

Customer email ID: partner501@company.com

Site Name

Technical Contact

Company Name

City

State/Province

Zip/Postal Code

Country

Phone

Following Generated Licenses have been sent to Email ID(s): partner501@company.com

ID Type	Unique Id	Transaction Key	Description	Status	License File
LID	pkeguceGFHM	A0D57D0038E39D9427131B	BR-NI-CES-2048-L3U	Success	License Key

[Generate Another License](#)

4. Download the license file to your PC by either clicking the hyperlink in the **License File** field, or saving the license file from the e-mail attachment.
5. Upload the license file to the Brocade device as instructed in [“Installing a license file”](#) on page 169.

Installing a license file

Once you obtain a license file, you can do one of the following:

- Copy the license file to a TFTP server to which the Brocade device has access and then use TFTP to copy the file to the license database of the ServerIron ADX switch.

- Copy the license file to the ServerIron ADX internal flash memory and then add the license to the ServerIron ADX license database.

Using TFTP to install a license file

To copy a license file from a TFTP server to the license database of the ServerIron ADX application switch, enter a command such as the following at the Privileged EXEC level of the CLI:

```
ServerIronADX# copy tftp license 10.1.1.1 lic.xml
```

Syntax: `copy tftp license <IP_address> <license_filename_on_host>`

The `<IP_address>` variable is the address of the IPv4 TFTP server.

The `<license_filename_on_host>` variable is the filename of the license file.

Copying a license to flash file system and loading to license database

To copy a license file to the flash file system of the ServerIron ADX application switch, for loading to the licence database, you must first copy the file to a TFTP server. To perform this copy operation, enter a command such as the following at the Privileged EXEC level of the CLI:

```
ServerIronADX# copy tftp usb0 10.1.1.1 ADX-1016-2-LIC-4P.xml tmp
```

Syntax: `copy tftp usb0 <IP_address> <license_filename_on_host> <filename_on_usb0>`

The `<IP_address>` variable is the address of the IPv4 TFTP server.

The `<license_filename_on_host>` variable is the filename of the license file.

The `<license_filename_on_usb0>` variable is the filename to be created on usb0.

To copy a license file from flash memory to the license database of the ServerIron ADX application switch, enter a command such as the following.

```
ServerIronADX# license add usb0/tmp
```

Syntax: `license add usb0/<filename_on_usb0>`

Verifying the license file installation

Use the **show license** command to verify that the license is installed on the device. Details about this command are in the section [“Viewing the license database”](#) on page 176.

Using a trial license

NOTE

A trial license must be ordered and installed by a Brocade representation only.

A trial license enables demonstration and evaluation of a licensed feature. The trial license is valid for a period of up to 45 days, and is renewable for an additional 45 days on the second time. A licensed feature operating under a trial license has the same functionality (CLI and **show** commands) as does a licensed feature operating under a normal license.

What happens when a trial license expires

A trial license will expire when it exceeds the specified expiration time or date. The countdown starts when the trial license is generated. When the license expires, the CLI commands related to the licensed feature will no longer be available from the CLI. The licensed feature will continue to run as configured until the software is reloaded, at which time the feature will be disabled and removed from the system.

NOTE

Trial licenses are not cumulative. The new license replaces the current license. To extend the license, you must contact your Brocade representative.

Console, and Syslog messages for trial license expiration

Three days prior to the date that a trial license is set to expire and each day until the last day, the following warning message will appear daily on the console. On the day that the license will expire, the warning message will appear every two hours. Syslog messages will also be generated.

```
SYSLOG: <12>Jan  1 00:00:00 License: Package ADX-1008-1-TRL-2PPLS with LID
GHIJWYDZNN expires in 3 days
```

When the license has expired, the following message will appear on the console. Syslog messages will also be generated.

```
SYSLOG: <13>Jan  1 00:00:00 License: Package ADX-1008-1-TRL-2PPLS with LID
GHIJWYDZNN has expired, use licence delete to remove it.
```

Renewing or extending a trial license

A trial license can be extended once by another trial license of the same type, or by a normal license of the same type. To avoid any interruptions to the network, obtain and install the second trial license before the first license expires. When extended by another trial license, the duration is not cumulative. The countdown starts when the trial license is generated.

To extend the license, you must contact your Brocade representative.

Deleting a license

A license will remain in the license database until it is deleted. To delete a license, enter a command such as the following at the Privileged EXEC level of the CLI:

```
ServerIronADX# license delete 7
```

NOTE

Only upgrade licences are allowed to be deleted. You cannot delete a base license.

This command immediately removes the license from the license database. The licensed feature will continue to run as configured until the software is reloaded, at which time the feature will be disabled and removed from the system. Syslog messages are generated when the license is deleted.

Syntax: `license delete <index_number>`

The `<index_number>` variable is a valid license index number. This information can be retrieved from the **show license** command output. For more information, refer to [“Viewing information about software licenses”](#) on page 174.

Viewing software license information from the Brocade software portal

This section describes other software licensing tasks supported from the Brocade software portal. You can use the **License Query** option to view software license information for a particular unit, transaction key, or both. You can export the report to Excel for sharing or archiving purposes.

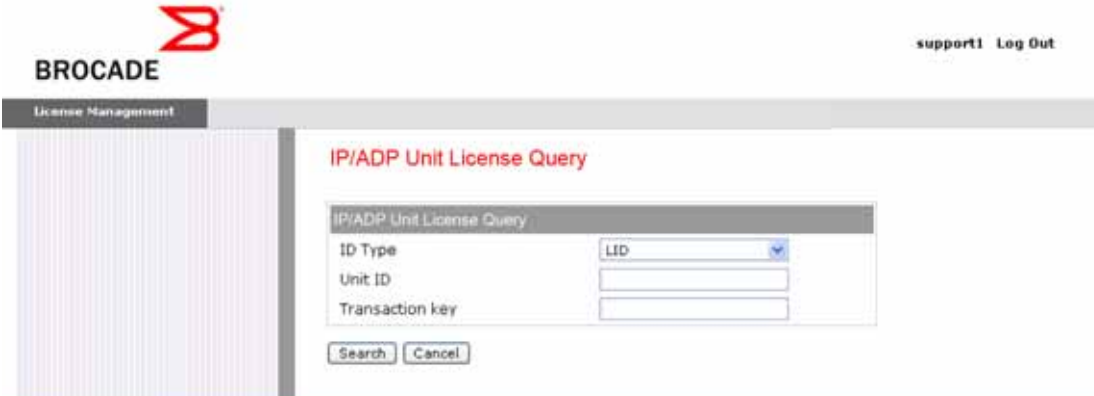
Depending on the status of the license, for example whether or not the license was generated, the report will include the following Information:

- Hardware part number, serial number, and description
- Software part number, serial number, and description
- Date the license was installed
- Transaction key
- LID
- Feature name
- Product line

From the **License Management** menu, select **Brocade IP/ADP > License Query**.

The **License Query** window displays. (Refer to Figure 9).

FIGURE 9 License Query window



The screenshot shows the Brocade IP/ADP Unit License Query window. At the top left is the Brocade logo. To the right of the logo is the text 'support1 Log Out'. Below the logo is a 'License Management' menu. The main content area is titled 'IP/ADP Unit License Query' in red. Below this title is a form titled 'IP/ADP Unit License Query' in a gray box. The form has three input fields: 'ID Type' with a dropdown menu showing 'LID', 'Unit ID', and 'Transaction key'. Below the form are 'Search' and 'Cancel' buttons.

- To view software license information for a particular unit, enter the LID in the **Unit ID** field and click **Search**.
- To view software license information for a particular transaction key, enter the unique number in the **Transaction key** field and click **Search**.

Figure 10 shows an example of the license query results.

FIGURE 10 License Query results window

BROCADE support@ Brocade Log Out

IP/ADP Unit License Query

IP/ADP Unit License Query

ID Type: LID

Unit ID:

Transaction key: F08D0E8B08842E13032C5D

Search Cancel

Level	Port Number	Serial Number	Installed Date	License File	Transaction Key	LID	Feature Name	Description	Product Line	Q
1	80-CER-2048C-AC	80CERAC1089				pkagrcSP00	NT CER 2048C 1	80CER2048C 1	80CER2048C	N
2	80-AC-CER-2048-ADU	UA200001757	02-Feb-2018	License File	F08D0E8B08842E13032C5D	pkagrcSP00	80-AC-CER-2048-ADU	80-AC-CER-2048-ADU	CER 2048C	N

Export to excel

Control UI | Site Map | Privacy | Legal | Investor Relations

In this example, the line items for Level 1 display hardware-related information and the line items for Level 2 display software-related information. If the query was performed before the transaction key was generated, the first row (Level 1) would not appear as part of the search results. Similarly, if the query was performed before the license was generated, some of the information in the second row would not be displayed.

Syslog messages information

The following Syslog messages are supported for software-based licensing.

TABLE 19 Syslog messages

Message level	Message	Explanation
Informational	License: Package <package_name> with LID <LID_number> is added	Indicates that the license package has been added.
Informational	License: Package <package_name> with LID <LID_number> is removed	Indicates that the license package has been deleted.
Warning	License: Package <package_name> with LID <LID_number> expires in <number> days	The trial license is about to expire. This message will begin to display 3 days before the expiration date, and every 2 hours on the last day that the license will expire.
Notification	License: Package <package_name> with LID <LID_number> has expired	The trial license has expired.

Viewing information about software licenses

This section describes the **show** commands associated with software licensing. These commands are issued on the Brocade device, at any level of the CLI.

Viewing the License ID (LID)

Brocade devices that ship during and after the release of software licensing will have the LID imprinted on the label affixed to the device. You also can use the CLI command **show version** to view the LID on these devices. Use the **show version** command to display the serial number, license, and LID of the device. This display differs for ServerIron ADX fixed configuration devices and ServerIron ADX chassis devices.

The following is example output from a ServerIron ADX fixed configuration device (ServerIron ADX 1000) with the license SI-1216-4-SSL installed.

```
ServerIronADX# show version
Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Boot Version 12.1.00T405 Oct 29 2009 10:12:19 PST label: dob12100
Monitor Version 12.1.00T405 Oct 29 2009 10:12:19 PST label: dob12100
System Version 12.0.00dT401 Mar 31 2010 10:59:57 PST label: ASM12000dev
AXP Version: 1.13 Dated: 2010/02/09 07:35:49
PAX Version: 0.0 Dated: 2009/07/28 10:35:11
MBRIDGE Version: 000b, Device ID # bebe
=====
Type:    ServerIronADX 1216-4
        Backplane Serial #:    SA98045987
        Chassis Serial #:      1234rtyu88
        LID #: GHIJWYDZNN License: SI-1216-4-SSL
        Part #:                84-1000811-01
        Version #:             13b626-03030303-111d8036-00
=====
```

For ServerIron ADX chassis devices, a separate license is displayed for the chassis and the ASM module. The following example displays the “**LID #**” for the chassis and the “**ASM-LID#**” for the ASM module.

```
ServerIronADX4000#show version
Copyright (c) 1996-2009 Brocade Communications Systems, Inc.
Boot Version 12.1.00aT405 Jul  9 2010 19:03:54 PDT label: dob12100a
Monitor Version 12.1.00aT405 Jul  9 2010 19:03:54 PDT label: dob12100a
System Version 12.3.01T401 May 26 2011 15:47:58 PDT label: ASM12301
AXP Version: 1.14 Dated: 2010/08/06 10:35:46
PAX Version: 8.4 Dated: 2010/10/06 14:31:02
MBRIDGE Version: 000b, Device ID # bebe

=====
Type:      ServerIron ADX 1216-4-PREM
          Backplane Serial #:   E20548F4EH
          Chassis Serial #:     E30549F00F
          LID #: gIFKJOhFFh License: SI-1216-4-SSL-PREM
          Part #: 35723-001
          SSL Card - Serial #: 1F0548F08W          Part #: 40-1000373-02
          Version #: 13b626-03030303-111d8036-00
=====
Active management module:
          1499 MHz Power PC processor (SVR 1.1, PVR 8021/0030) 599 MHz bus
          512 KB Boot flash
          131072 KB Code flash
          2048 MB DRAM
The system uptime is 14 days 4 hours 2 minutes 59 seconds
```

Viewing the license database

To display general information about all software licenses in the license database, use the **show license** command. The following shows example output.

```
ServerIronADX#show license
```

Index	Package Name	Lid	Type	Status	Period
1	SI-1008-1	GHIJWYDZNN	Normal	Active	Unlimited
2	ADX-1008-1-LIC-2PPLS	GHIJWYDZNN	Normal	Active	Unlimited
3	ADX-1016-2-LIC-4P	GHIJWYDZNN	Normal	Active	Unlimited
4	ADX-1016-4-LIC-10G	GHIJWYDZNN	Normal	Active	Unlimited

To display information about Application Switch Module (ASM) licenses, use the **show license asm** command. The following shows example output.

Chassis with only one ASM4 on the box in slot 1:

```
ServerIronADX# show license asm
```

Slot 1:

Index	Package Name	LID	License Type	Status	License Period
1	SI-ASM4	uCOjkwenb	normal	active	unlimited

Output after upgrading to an ASM8 in slot 1:

```
ServerIronADX# show license asm
```

Slot 1:

Index	Package Name	LID	License Type	Status	License Period
1	SI-ASM4	uCOjkwenb	normal	active	unlimited
2	ADX-CH-LIC-ASM4-8	uCOjkwenb	normal	active	unlimited

To display detailed information about a particular license, use the **show license <index_number>** command. The following shows example output.

```
ServerIronADX# show license 1
```

License information for license <1>:

```
+package name: SI-1008-1
+lid: GHIJWYDZNN
+Status: Active
+license type: Normal
+license period: Unlimited
```

Syntax: **show license [asm] [<index_number>]**

The following table describes the information displayed by the **show license** command.

TABLE 20 Output from the **show license** command

This field...	Displays...
Index	The license hash number that uniquely identifies the license.
Package Name	The package name for the license.
LID	The license ID. This number is embedded in the Brocade device.
Status	<p>Indicates the status of the license:</p> <ul style="list-style-type: none"> • Invalid – A license is invalid when the LID doesn't match the serial number of the device for which the license was purchased. • Active – The license is valid and in effect on the device. • Expired – For trial licenses only, this indicates that the trial license has expired. <p>This field will always display "active" with the show license asm command.</p>
Type	Indicates whether the license is normal (permanent) or trial (temporary). This field will always display "normal" with the show license asm command.
Period	If the license type is trial (temporary), this field will display the number of days the license is valid. If the license type is normal (permanent), this field will display "unlimited". Because trial licenses are not available for the ASM4 or ASM4 to ASM8 upgrade, this field will always display "unlimited" with the show license asm command.

License precedence

This field displays in the output of the **show license <license_index_number>** command. It indicates <<what?>>

Trial license information

The following details display in the output of the **show license <index_number>** command.

+ days used	The number of days the trial license has been in effect.
+ hours used	The number of hours the trial license has been in effect.
+ days left	The number of days left before the trial license expires.
+ hours left	The number of hours left before the trial license expires.

Brocade license information

The following details display in the output of the **show license <index_number>** command.

+ pkg info	The information in this field is used by engineering and technical support staff for debugging purposes.
------------	--

5 Viewing information about software licenses